

A Stable Self-Structuring Adaptive Fuzzy Control Scheme for Continuous Single-Input Single-Output Nonlinear Systems

by

Phi Anh Phan
B.Eng. (Mechatronics Eng., Hons.)

Submitted in fulfilment of the requirements for the Degree of Doctor of Philosophy

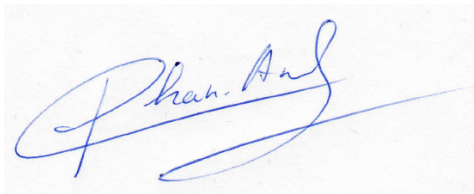


**School of Engineering, University of Tasmania
March 2009**

Statement of Originality & Authority of Access

This thesis contains no material which has been accepted for a degree or diploma by the University of Tasmania or any other institution, except by way of background information and has been duly acknowledged in this thesis, and to the best of the author's knowledge and belief no material has previously been published or written by another person except where due acknowledgement is made in the text of this thesis.

This thesis contains confidential information and is not to be disclosed or made available for loan or copy without the expressed permission of the University of Tasmania. Once released the thesis may be made available for loan and limited copying in accordance with the Copyright Act 1968.

A handwritten signature in blue ink, reading "Phan. Anh". The signature is written in a cursive style with a long horizontal stroke extending to the right.

School of Engineering,
University of Tasmania,
Hobart, Tasmania, Australia

Abstract

Adaptive fuzzy control has been an active research area in the past decade. Fundamental issues such as stability, robustness, and performance analysis have been solved. However, one main drawback is the generally fixed structure of the fuzzy controllers, which are normally chosen by trial-and-error in practice. Few attempts to develop self-structuring AFC have been reported, and important issues such as stability, computational efficiency, and implementability have not been investigated thoroughly. In particular, the stability of the system when the structure changes has not been proven. Thus, a more effective self-structuring AFC scheme is desirable.

The main objective of the research is to develop a stable self-structuring AFC scheme for continuous-time single-input-single-output (SISO) uncertain nonlinear systems.

A novel online self-structuring adaptive fuzzy control scheme that is applicable for a number of classes of continuous SISO nonlinear systems is proposed. The applicable classes include affine nonlinear systems, non-affine nonlinear systems, and nonlinear systems in triangular forms. The main features of the proposed control scheme are:

- It needs less restriction on the controlled plants and no restriction on the design parameters.
- It employs a modified adaptive law that guarantees *explicit* boundedness of adaptive parameters and control action.
- The self-structuring algorithm is relatively simple and guarantees explicit boundedness of the number of rules generated.
- Only triangular membership functions are generated and only 2 membership functions are allowed to overlap to increase the interpretability of generated fuzzy controllers.
- High-gain observers are used when not all the states are measurable and the design of observers is completely separated from the design of controllers.
- For nonlinear systems in triangular forms, only one fuzzy system is needed (unlike the back-stepping approach where one fuzzy system is needed at each step).

- An approximation error estimator and an automatic switching mechanism can be used to further increase the robustness and computational efficiency.

The stability of the overall system, especially when the structure changes, is guaranteed using the Lyapunov stability technique. The overall system is stable in the sense that all the variables are bounded (including number of rules generated) and the tracking error is uniformly ultimately bounded. The proposed control algorithms are implemented in Matlab and Simulink for ease of simulation and practical application. Numerous simulation examples are performed to demonstrate the theoretical results.

The proposed control scheme makes practical application of AFC easier. Designers need to specify only a few design parameters and no longer have to specify the controller structure by trial and error. A simulation or application can be quickly and easily implemented using the developed controllers in Simulink.

Publications

As part of this research, the following papers have been published:

Journal papers:

- P.A. Phan, and T.J. Gale, "Two-mode adaptive fuzzy control with approximation error estimator", *IEEE Transactions on Fuzzy Systems*, volume 15 (5), pp 943-955, Oct. 2007.
- P.A. Phan, and T.J. Gale, "Direct adaptive fuzzy control with less restrictions on the control gain", *International Journal of Control, Automation, and Systems*, Vol 5 No 6 Dec 2007, in press.
- P.A. Phan, and T.J. Gale, "Direct Adaptive Fuzzy Control with a Self-Structuring Algorithm", *Fuzzy Sets and Systems*, in press.

Conference papers:

- P. A Phan, and T.J. Gale, 'Stable Adaptive Fuzzy Controller with Approximation Error Estimators', *Proceedings of The 2nd International Conference on Artificial Intelligence in Science and Technology*, Hobart, Tas. Australia, 295-300 (2004)
- P.A. Phan, and T.J. Gale, "Adaptive fuzzy control: approximation error estimator approach", *23rd IASTED International Conference on Artificial Intelligence and Applications*, Innsbruck, Austria, 2005.
- P.A. Phan, and T.J. Gale, "Direct adaptive fuzzy control with less restrictions on the control gain", *International Conference on Computational Intelligence for Modelling, Control, and Automation*, Sydney, Australia, 2006.

Acknowledgements

I would like to thank my supervisor, Dr. Timothy J Gale, for his endless guidance, support and friendship throughout the duration of the research. He was always there whenever I needed help.

I would also like to thank my associate supervisors, Dr. Bernardo A Leo de La Barra and Prof Michael Negnevitsky, for their advices and help in Control and Artificial Intelligence.

Most importantly, I would like to thank my family for their love and support. Without them, I would not have completed this project.

Finally, I would like to thank all my friends who have helped and assisted me during my time at University of Tasmania.

Table of Contents

ABBREVIATIONS.....	9
1. CHAPTER 1 INRODUCTION	10
1.1. INTRODUCTION	10
1.2. ADAPTIVE FUZZY CONTROL	10
1.2.1. <i>What is adaptive fuzzy control?</i>	11
1.2.2. <i>Why adaptive fuzzy control?</i>	11
1.2.3. <i>Relationship between adaptive fuzzy control and adaptive neural network control</i>	13
1.3. MOTIVATION AND OBJECTIVES	13
1.4. OUTLINE OF THE THESIS	14
1.5. CONCLUSION	16
2. CHAPTER 2 GENERAL LITERATURE REVIEW AND PRELIMINARIES.....	17
2.1. INTRODUCTION	17
2.2. A REVIEW ABOUT THE DEVELOPMENT OF ADAPTIVE FUZZY CONTROL	17
2.2.1. <i>Structure</i>	17
2.2.1.1. Direct AFC	17
2.2.1.2. Indirect AFC	18
2.2.1.3. AFC combined with other controllers	18
2.2.2. <i>Different classes of nonlinear systems</i>	20
2.2.2.1. Affine and non-affine nonlinear systems	20
2.2.2.2. Strict-feedback and pure-feedback nonlinear systems	22
2.2.2.3. SISO and MIMO nonlinear systems	24
2.2.2.4. State-feedback and output feedback nonlinear systems	25
2.2.2.5. Continuous and discrete systems	25
2.2.3. <i>Adaptive mechanism of fuzzy systems</i>	26
2.2.3.1. Only parameters are tuned	26
2.2.3.2. Both parameters and structure are adjusted	26
2.3. PRELIMINARIES.....	27
2.3.1. <i>Fuzzy system and neural network</i>	27
2.3.2. <i>Concepts of stability and boundedness</i>	27
2.3.2.1. Stability definitions	28
2.3.2.2. Boundedness definitions.....	29
2.3.3. <i>Lyapunov stability theorem</i>	29
2.3.3.1. Conditions for stability	30
2.3.3.2. Conditions for boundedness	31
2.3.4. <i>Universal approximation properties</i>	31
2.3.4.1. Universal approximation property for zero-order Takagi-Sugeno fuzzy systems.....	31
2.4. BASIC INDIRECT ADAPTIVE FUZZY CONTROL FOR SISO AFFINE NONLINEAR SYSTEMS.....	33
2.5. CONCLUSION	37
3. CHAPTER 3 TWO-MODE INDIRECT ADAPTIVE FUZZY CONTROL WITH APPROXIMATION ERROR ESTIMATOR.....	39
3.1. INTRODUCTION	39
3.2. LITERATURE REVIEW	39
3.3. TWO-MODE ADAPTIVE FUZZY CONTROL WITH APPROXIMATION ERROR ESTIMATOR.....	40
3.4. APPLICATIONS	45
3.4.1. <i>Control of an inverted pendulum</i>	45
3.4.2. <i>Control of a Chua's chaotic circuit</i>	47
3.5. CONCLUSION	49
4. CHAPTER 4 DIRECT ADAPTIVE FUZZY CONTROL WITH LESS RESTRICTION ON THE CONTROL GAIN	54
4.1. INTRODUCTION	54
4.2. LITERATURE REVIEW	54
4.3. DIRECT ADAPTIVE FUZZY CONTROL WITH LESS RESTRICTION	56
4.4. APPLICATIONS	62
4.4.1. <i>Inverted pendulum</i>	62

4.4.2.	Magnetic levitation system.....	63
4.5.	CONCLUSION	64
5.	CHAPTER 5 SELF-STRUCTURING DIRECT ADAPTIVE FUZZY CONTROL	69
5.1.	INTRODUCTION	69
5.2.	LITERATURE REVIEW	69
5.3.	SELF-STRUCTURING DIRECT ADAPTIVE FUZZY CONTROL FOR AFFINE NONLINEAR SYSTEMS.....	71
5.3.1.	Description of the self-structuring algorithm.....	72
5.3.1.1.	Criteria for rule generation	73
5.3.1.2.	Adding a membership function and its related rules when the \mathcal{E} -completeness is not satisfied	73
5.3.1.3.	Replacing a membership function and its related rules when the \mathcal{E} -completeness is not satisfied	75
5.3.1.4.	Adding a membership function and its related rules when $\underline{e}^T P \underline{b}_C$ is equal to or larger than <i>error _ threshold</i>	76
5.3.1.5.	Replacing a membership function and its related rules when the error measurement $\underline{e}^T P \underline{b}_C$ is equal to or larger than <i>error _ threshold</i>	78
5.3.1.6.	Parameters	79
5.3.2.	SSDAFC.....	79
5.3.2.1.	When the structure is fixed.....	80
5.3.2.2.	When the structure changes	81
5.4.	EXAMPLES	82
5.4.1.	Inverted pendulum.....	82
5.4.2.	Magnetic levitation	84
5.5.	CONCLUSION	85
6.	CHAPTER 6 SELF-STRUCTURING DIRECT ADAPTIVE FUZZY CONTROL FOR NON-AFFINE NONLINEAR SYSTEMS.....	90
6.1.	INTRODUCTION	90
6.2.	LITERATURE REVIEW	90
6.3.	SSDAFC FOR NON-AFFINE NONLINEAR SYSTEMS.....	92
6.3.1.	Existence of an ideal control law.....	92
6.3.2.	Stability analysis	93
6.4.	EXAMPLES	95
6.4.1.	Application 1	95
6.4.2.	Application 2	97
6.5.	CONCLUSION	98
7.	CHAPTER 7 EXTENSION TO THE CONTROL OF OTHER CLASSES OF SISO NON-AFFINE NONLINEAR SYSTEMS.....	105
7.1.	INTRODUCTION	105
7.2.	SSDAFC OF SYSTEMS IN THE FORM (7.2)	106
7.2.1.	Control of system (7.2) with strong relative degree $\rho = n$	107
7.2.2.	Control of system (7.2) with strong relative degree $\rho < n$	107
7.3.	SSDAFC OF SYSTEMS IN THE TRIANGULAR FORM (7.3)	108
7.4.	OUTPUT FEEDBACK SSDAFC.....	111
7.5.	EXAMPLE.....	117
7.5.1.	Continuously stirred tank reactor (CSTR) system without zero dynamics	117
7.5.2.	Continuously stirred tank reactor (CSTR) system with zero dynamics	120
7.5.3.	Third-order system in triangular form (7.3).....	122
7.6.	CONCLUSION	124
8.	CHAPTER 8 MATLAB IMPLEMENTATION	133
8.1.	INTRODUCTION	133
8.2.	PROGRAMMING.....	133
8.3.	ADAPTIVE FUZZY CONTROL SIMULINK LIBRARY	134
8.3.1.	DAFC block	134
8.3.2.	SSDAFC block	136

8.3.3.	<i>High-gain observer block</i>	137
8.4.	AFC SIMULATION.....	138
8.4.1.	<i>Create a Simulink model for the AFC application</i>	138
8.4.2.	<i>Design the fuzzy system that is used as the initial controller</i>	139
8.4.3.	<i>Load the controller's parameters</i>	140
8.4.4.	<i>Perform simulation</i>	140
8.5.	REAL-TIME AFC.....	141
8.6.	CONCLUSION.....	143
9.	CHAPTER 9 DISCUSSION AND CONCLUSION	144
9.1.	DISCUSSION.....	144
9.1.1.	<i>Main contributions</i>	144
9.1.2.	<i>Limitations</i>	145
9.1.3.	<i>Future research</i>	145
9.2.	CONCLUSION.....	146
	APPENDIXES	147
	APPENDIX 3.A.....	147
	APPENDIX 3.B.....	151
	APPENDIX 4.B.....	152
	APPENDIX 6.A.....	154
	REFERENCES	156

Abbreviations

AFC: Adaptive Fuzzy Control

AIC: Adaptive Intelligent Control

ANNC: Adaptive Neural Network Control

CSTR: Continuous Stirred Tank Reactor

FLC: Fuzzy Logic Controller

GUI: Graphical User Interface

MIMO: Multi Input Multi Output

NN: Neural Network

SISO: Single Input Single Output

SSAFC: Self-Structuring Adaptive Fuzzy Control

SSDAFC: Self-Structuring Adaptive Fuzzy Control

UUB: Uniform Ultimate Boundedness or Uniformly Ultimately Bounded

1. Chapter 1

INTRODUCTION

1.1. Introduction

This chapter introduces the thesis and adaptive fuzzy control (AFC), giving a formal definition of AFC and its advantages, then the motivations and objectives of the research. Finally, the outline of the thesis is given, including how the thesis will be organised, what will be presented in each chapter and how they are linked together.

1.2. Adaptive fuzzy control

The early 1990s have witnessed a rapid growth of successful applications of fuzzy logic to automatic control. Examples of such applications are washing machines, electronically stabilized camcorders, auto-focus cameras, air conditioners, automobile transmissions, and subway trains [1]. Indeed, Fuzzy Logic Controllers (FLCs) offer an alternative to the control of complex nonlinear systems that are not easily controlled by conventional automatic control methods as they provide a framework to incorporate linguistic fuzzy information from human experts while not requiring a mathematical model of the plant. However, there is lack of mathematical analysis of stability, robustness, and systematic design procedure. This substantially restricts the application domain of FLCs.

On the other hand, adaptive control has a long history of intense activities involving stability proof, robustness design, and performance analysis [2]. The advances in stability theory and the progress of control theory in the 1960s have improved the understanding of adaptive control. In the mid 1980s, research of adaptive control mainly focused on robustness in the presence of unmodeled dynamics and bounded disturbances. Motivated by the early success of adaptive control of linear systems, the extension to nonlinear systems has been investigated from the end of 1980s to early 1990s. Thus, adaptive control offers powerful mathematical tools to the analysis of stability and robustness of nonlinear control systems.

Thus, it is logical to think that combining fuzzy control and adaptive control may give a better control methodology. The result is adaptive fuzzy control (AFC). Understandably, AFC possesses the advantages of both methodologies. It has the linguistic knowledge representability and parallel computing of fuzzy systems, and

the stability and robustness of conventional adaptive controllers. Formal definition of adaptive fuzzy control is given next.

1.2.1. What is adaptive fuzzy control?

Wang defines an adaptive fuzzy system as a fuzzy logic system equipped with a training algorithm, where the training algorithm adjusts the parameters (and the structures) of the fuzzy logic system based on numerical information. According to this definition, neuro-fuzzy systems, in which fuzzy systems are represented by neural networks, are also adaptive fuzzy systems.

An adaptive fuzzy controller can be defined as a controller, in which adaptive fuzzy systems are employed and adaptive control theory is used to derive training algorithms such that stability and performance of the closed-loop system are guaranteed.

Lyapunov stability techniques play a critical role in the design and stability analysis of the adaptive systems [2]. A Lyapunov function candidate is a mathematical function designed to provide a simplified scalar measure of the control objectives. The control objectives are met when the chosen Lyapunov function is driven to zero. More details about Lyapunov stability are given in chapter 2. In adaptive fuzzy control systems, stability is investigated by studying the behaviour of some Lyapunov function candidates.

In summary, a controller is called an adaptive fuzzy controller if it possesses both of the following features:

- Adaptive fuzzy systems are employed
- Lyapunov stability technique is used to derive training algorithms to guarantee the stability of the closed-loop system.

1.2.2. Why adaptive fuzzy control?

The advantage of AFC, combining both fuzzy control and adaptive control, includes the followings.

- Fuzzy control allows incorporating linguistic fuzzy information from human operators. The operators can describe how they control the system under control (or how the system behaves) in term of fuzzy If-Then rules. This information is easily captured by fuzzy systems.

- Fuzzy control provides universal nonlinear approximators. Fuzzy systems are nonlinear universal approximators. In conventional linear robust adaptive control studies, linear approximators are used to approximate some unknown functions that are assumed to be linear. Using fuzzy systems in adaptive control relaxes the assumption that the unknown function must be linear. Thus, it provides an extension to create nonlinear robust control schemes where there is no need to assume that the plant is a linear parameterization of *known* nonlinear functions [3].
- Fuzzy control is easy to understand. Because fuzzy control emulates human control strategy, its principle is easy to understand for noncontrol specialists. During the past two decades, conventional control theory has been using more and more advanced mathematical tools. This results in fewer and fewer practical engineers who can understand the theory. Therefore, practical engineers tend to use approaches which are simple and easy to understand. Fuzzy control is such an approach [1].
- Fuzzy control is simple to implement. Fuzzy logic systems, which are the heart of fuzzy control, possesses a high degree of parallel implementation. Many fuzzy VLSI chips have been developed, which make the implementation of fuzzy controllers simple and fast.
- Fuzzy control is cheap to develop. Because fuzzy control is easy to understand and simple to implement, the software and hardware cost is low. Also, there are a wide range of software tools available for designing fuzzy controllers (e.g. Matlab).
- Adaptive control is a model-free approach. It does not require a mathematical model of the system. Adaptive algorithms are used to adjust the parameters *online* in such a way that the control objectives are met. Thus, a mathematical model of the plant is not needed.
- Adaptive control guarantees stability and robustness. Stability and robustness are the most important issues in control theory. Stability means that for any bounded input over any amount of time, the output will also be bounded. Robustness refers to the ability of the control system to maintain stability even in the presence of unmodeled dynamics or external disturbances. Traditional fuzzy control cannot guarantee stability and robustness of the control system. In

adaptive control, Lyapunov stability technique provides the mathematical framework to establish adaptive algorithms that guarantee stability and robustness.

- Adaptive control provides a systematic design approach. There is no standard systematic design procedure in traditional fuzzy control. The tuning of parameters is mostly based on trial and error approach. Thus, it is a time consuming and ill-defined process. Adaptive control provides a systematic design approach, in which parameters and adaptive laws can be chosen explicitly using Lyapunov technique.

1.2.3. Relationship between adaptive fuzzy control and adaptive neural network control

Adaptive neural network control (ANNC) is a control method, in which neural networks are employed and adaptive control theory is used to derive training algorithms such that stability and performance of the closed-loop system are guaranteed. Thus, compared to AFC, the main difference is that neural networks are used, instead of fuzzy systems, as approximators.

Moreover, it is well known that a fuzzy system can be realized by a neural network. Many ANNC schemes can be converted to AFC schemes and vice versa. Therefore, it would be inadequate to survey only AFC and ignore ANNC.

In subsequent chapters, ANNC is also considered and is mentioned when it is relevant. The term “adaptive intelligent control” (AIC) will be used to refer to both AFC and ANNC.

1.3. Motivation and Objectives

With the advantages mentioned above, AFC is a very good candidate for control of uncertain nonlinear dynamic systems. However, there are still some drawbacks that obstruct the practical application of AFC.

One main drawback is the generally fixed structure of the fuzzy controllers, which are normally chosen by trial-and-error in practice. Few attempts to develop self-structuring AFC have been reported, and important issues such as stability, computational efficiency, and implementability have not been investigated thoroughly. In particular, the stability of the system when the structure changes has not been proven. Thus, a more effective self-structuring AFC scheme is desirable.

Other drawbacks include restrictions on the classes of applicable nonlinear systems, constraints on the design parameters that are hard to determine in practice, the complexity of controllers for nonlinear systems in triangular forms, etc.

With the desire to make AFC easier for practical application, the objectives are as follows.

Objectives:

- i. Develop a novel online self-structuring AFC scheme that is applicable for a wide range of continuous SISO nonlinear systems.
- ii. Propose solutions to overcome drawbacks such as:
 - Improve computational efficiency by proposing 2-mode adaptive fuzzy control
 - Relax the extra restrictions of the direct adaptive fuzzy control
 - Reduce the complexity of the control of nonlinear systems in triangular form
- iii. Develop implementation software in order to make simulation and practical application of the proposed AFC scheme fast and easy.

To achieve these objectives, the rest of the thesis is carried as follows.

1.4. Outline of the thesis

Chapter 2 provides a general literature review and mathematical preliminaries. First, we give a brief survey about the development of AFC. Then, some required mathematical preliminaries are given. Finally, basic concepts of AFC (such as *ideal control*, *minimum approximation error*, *ideal parameters*, etc. and how the stability analysis and adaptive laws are derived using Lyapunov stability theorem) are introduced through a simple AFC scheme, basic indirect adaptive fuzzy control for affine nonlinear systems. The shortcomings of this basic AFC scheme are also discussed.

In addition to a general literature review in chapter, there is a separate literature review for each major topic (chapters 3, 4, 5, 6, 7).

One shortcoming of basic AFC is the effect of the approximation error, which can de-stabilize the closed-loop system. In chapter 3, we propose a novel 2-mode indirect AFC scheme, in which an approximation error estimator is used to compensate for the approximation error. Moreover, the control scheme can switch between learning mode and operating mode using a switching mechanism. The

switching mechanism improves the computational efficiency in cases where the controlled plants satisfy certain conditions.

Direct AFC is simpler than indirect AFC but it normally requires more restrictions on the control gain than the indirect one. This limits the application of direct AFC in practice. In chapter 4, we propose a direct AFC scheme with less restriction. By using an extension of the approximation theorem, we show that direct AFC actually requires the same restrictions as the indirect one. Also, the proposed control scheme employs a modified adaptive law that guarantees explicit boundedness of adaptive parameters and control action.

In chapter 5, based on the direct AFC scheme proposed in chapter 4, we propose a self-structuring direct AFC scheme for SISO affine nonlinear systems. Compared to some existing algorithms, the proposed self-structuring algorithm is relatively simpler and also guarantees explicit boundedness of the number of rules generated. Only triangular membership functions are generated and only 2 membership functions are allowed to overlap to increase the interpretability of generated fuzzy controllers.

In chapter 6, we extend the result of chapter 5 to a class of non-affine nonlinear systems. By using the implicit function theorem and an extension of the approximation theorem, we show that the AFC scheme proposed in chapter 5 can also be applied to non-affine nonlinear systems.

In chapter 7, we further extend the result to larger classes of nonlinear systems. By using the concepts of Lie derivative and strong relative, a wider class of non-affine nonlinear systems and a class of nonlinear systems in triangular systems can be transformed to the form in chapter 6. Thus, the AFC scheme proposed in chapter 5 can also be applied to these classes of nonlinear systems. For the class of nonlinear systems in triangular systems, this approach requires only one fuzzy system (unlike the back-stepping approach where one fuzzy system is needed at each step). The approach requires the output and its derivatives, which sometimes are not available for measurement. In this case, high-gain observers are proposed to estimate the derivatives. The design of observers is completely separated from the design of controllers.

In chapter 8, the software implementation of the proposed control algorithms is presented. Using Mathworks, we develop a self-structuring AFC library, which includes some control blocks that are ready to be used. By simple click-and-drag

mouse operations, a simulation or real-time application of self-structuring AFC can be performed quickly and easily.

Chapter 9 presents discussion and conclusions.

1.5. Conclusion

An introduction to the thesis is given in this chapter. The main objectives of the thesis are to develop a novel online self-structuring AFC scheme, improve results of existing AFC schemes, and to develop software to implement the developed AFC scheme.

2. Chapter 2

GENERAL LITERATURE REVIEW AND PRELIMINARIES

2.1. Introduction

This chapter provides background for the thesis. First, a review is presented in section 2.2 to give a general picture about the development of AFC in the past decade. Then, important mathematical background such as stability concept and Lyapunov stability technique is presented in section 2.3. Finally, the basic framework of AFC is introduced in section 2.4 through a simple example of indirect AFC of affine nonlinear systems.

2.2. A review about the development of adaptive fuzzy control

From the early 1990s, adaptive fuzzy control has been an active research area. Many researchers have contributed their work to the field. A great number of different control approaches, methods, schemes, and control applications have been published in various books, journals, and conferences. Thus, providing a complete description of adaptive fuzzy control in a single context is impossible. In this section, a brief review is given in order to demonstrate the wide range of adaptive fuzzy control schemes available in the literature, from different configuration structures, applicable classes of nonlinear systems, to adaptive mechanisms of fuzzy systems.

2.2.1. Structure

In their simplest forms, adaptive fuzzy controllers are constructed only by adaptive fuzzy systems. They can be classified into two categories: direct and indirect adaptive fuzzy control.

2.2.1.1. Direct AFC

Direct adaptive fuzzy controllers use adaptive fuzzy systems as controllers [1]. The adaptive mechanism is then designed to adjust the adaptive fuzzy system in such a way that will stabilize the plant and make the closed-loop system achieve its performance objectives. Direct adaptive fuzzy controllers have been proposed in [1, 4, 5].

2.2.1.2. Indirect AFC

Unlike direct adaptive fuzzy controllers, indirect adaptive fuzzy controllers use adaptive fuzzy logic systems to model the plant and construct the controllers assuming that the fuzzy logic systems represent the true plant. Indirect adaptive fuzzy controllers have been presented in [4-9].

2.2.1.3. AFC combined with other controllers

Pure direct and indirect adaptive fuzzy controls are simple, but they also have disadvantages. Thus, in the later years, it is often that adaptive fuzzy control is combined with other control techniques.

- Direct AFC combined with indirect AFC: [10-13] propose hybrid direct and indirect adaptive fuzzy control schemes in which the control output is the weighted average of a direct and an indirect adaptive fuzzy controllers. This combination provides a framework to incorporate both linguistic knowledge describing the plant behaviour and the control actions.
- AFC combined with another controller to compensate for approximation error: In general, there exist approximation errors when approximating nonlinear functions by fuzzy systems. These approximation errors may effect and deteriorate the stability and performance of adaptive fuzzy control systems. To overcome this problem, previous researchers have proposed combining AFC with another controller. [14] proposes a control scheme in which an indirect adaptive fuzzy controller is combined with a fuzzy sliding mode controller. The fuzzy sliding mode controller is designed to compensate for the approximation errors. [15-20] propose adaptive fuzzy control with a variable structure control term. The variable structure control term is designed using some known bounds of approximation errors. The term is then added to the control output to compensate for the effect of approximation errors. However, the bounds of approximation errors are normally hard to obtain in practice. Thus, they take a step further by proposing some adaptive mechanisms to estimate these bounds online [21, 22].
- AFC combined with output feedback control: In many applications, it is impossible or too expensive to measure all the state variables of the system under control. Output feedback control is an approach to overcome this

difficulty. The only variable needed to be measured is output of the system. Many adaptive fuzzy control schemes based on output feedback control have been proposed in the literature: [16, 23].

- AFC combined with H_∞ control: External disturbances play an important role in real control applications. They not only deteriorate control performance but also may cause instability. H_∞ optimal control is a technique used in traditional control theory to minimize the effect of external disturbances. [24-30] use adaptive fuzzy control combined with H_∞ control technique to attenuate the effect of disturbances.
- AFC combined with a supervisory control: An adaptive fuzzy controller sometime does not adapt fast enough. It leads to the state variables of the controlled system moving outside of a desired constraint set. This problem can be solved by increasing adaptive gains. However, adaptive gains cannot be too large. Increasing adaptive gains increases sensitivity to noise, leading to chattering of control output. Thus, to keep the state variables of the system under control in a desired constraint set without the need of large adaptive gains, some researchers [1, 13, 31, 32] propose adaptive fuzzy control combined with a supervisory control. This supervisory control is also a variable structure control term, which is designed using knowledge of the bounds of the unknown nonlinear functions. When the state variables are well inside the constraint set, the supervisory control is zero. When the state variables tend to move outside of the desired boundaries, the supervisory control begins to operate to force the states to stay in the constraint set.
- AFC combined with more than one other control techniques: [12] proposes an adaptive fuzzy control scheme, in which the control output is a combination of a direct adaptive fuzzy controller, an indirect adaptive fuzzy controller, and a variable structure control term to compensate for approximation errors. The bounds used in the variable structure control term are estimated online, thus no prior knowledge about the bounds is required. [13] proposes hybrid direct and indirect adaptive fuzzy control with a supervisory controller. [17] proposes an adaptive fuzzy control scheme combined with variable structure control and H_∞ control such that

both the effects of approximation errors and external disturbances can be attenuated to any prescribed level.

In general, adaptive fuzzy control combined with other control schemes overcome disadvantages existing in pure direct and indirect adaptive fuzzy control. However, they are more complicated in both theoretical analysis and implementation. Thus, for a particular application, it is up to control designers to decide when it is necessary to combine adaptive fuzzy control with another control technique.

2.2.2. Different classes of nonlinear systems

In the theory of nonlinear control, the control of different classes of nonlinear systems has been considered. Different classes of nonlinear systems have different characteristics, and thus require different control techniques. Some well-established techniques are available for different classes of nonlinear systems. For example, linearizable nonlinear systems can be treated using feedback linearization techniques. Nonlinear systems in strict-feedback forms can be treated using backstepping design. Nonlinear systems, in which not all the state variables are measurable, can be dealt with using output feedback control, etc. These results in nonlinear control have inspired researchers to propose a number of adaptive fuzzy control schemes for these classes of nonlinear systems based on the available techniques.

Here, we review AFC schemes in terms of the nonlinear classes that they can be applied to.

2.2.2.1. Affine and non-affine nonlinear systems

- Affine nonlinear systems

Under some geometric conditions, the input-output response of a class of single input-single-output (SISO) nonlinear systems can be rendered to the following *Brunovsky form* [2]:

$$\begin{cases} \dot{x}_i = x_{i+1}, i = 1, \dots, n-1 \\ \dot{x}_n = f(\underline{x}) + g(\underline{x})u + d(t) \\ y = x_1 \end{cases} \quad (2.1)$$

where $\underline{x} = [x_1, x_2, \dots, x_n]^T \in R^n$, $u \in R$, $y \in R$ are the state variables, system input and output, respectively; $f(\underline{x})$ and $g(\underline{x})$ are smooth functions; and $d(t)$ denotes the external disturbance bounded by a known constant $d_0 > 0$, i.e. $|d(t)| \leq d_0$.

Nonlinear systems that can be represented in this form are also known as affine nonlinear systems as the systems are linear in the input variables.

If $f(\underline{x})$ and $g(\underline{x})$ are known, the feedback linearization technique can be used to design a controller. The most common control structure is

$$u = \frac{1}{g(\underline{x})} [-f(\underline{x}) + v] \quad (2.2)$$

where v is a new control variable. In cases where $f(\underline{x})$ and $g(\underline{x})$ are unknown, adaptive fuzzy control has been proposed.

[1, 4-7] propose indirect adaptive fuzzy control schemes for affine nonlinear systems, in which two adaptive fuzzy systems $\hat{f}(\underline{x}|\underline{\theta}_f)$ and $\hat{g}(\underline{x}|\underline{\theta}_g)$ are used to approximate $f(\underline{x})$ and $g(\underline{x})$ respectively. Lypapunov stability analysis is used to derive the adaptive laws and to guarantee the control objectives. In these approaches, it should be noted that additional precautions are required to avoid possible singularities of the controllers (i.e., $\hat{g}(\underline{x}|\underline{\theta}_g) = 0$). For instance, in Wang [1], a projection algorithm is proposed for adjusting $\underline{\theta}_g$ to avoid singularities.

[24, 25, 32] propose direct adaptive fuzzy control schemes for nonlinear affine systems. In these schemes, only one adaptive fuzzy system $\hat{u}(\underline{x}, v|\underline{\theta})$ is used to approximate the control $u = \frac{1}{g(\underline{x})} [-f(\underline{x}) + v]$. Direct adaptive fuzzy control schemes avoid control singularity problem completely. However, compared to indirect schemes, more restrictions on $g(\underline{x})$ are normally required. More discussion on the restrictions of direct AFC will be given in chapter 4.

- Non-affine nonlinear systems

Non-affine nonlinear systems is a broader class of nonlinear systems, whose input variables may not be expressed in an affine form. A SISO non-affine nonlinear system is defined as:

$$\begin{cases} \dot{x}_i = x_{i+1}, i = 1, \dots, n-1 \\ \dot{x}_n = f(\underline{x}, u) \\ y = x_1 \end{cases} \quad (2.3)$$

where $\underline{x} = [x_1, x_2, \dots, x_n]^T \in R^n$, $u \in R$, $y \in R$ are the state variables, system input and output, respectively; $f(\underline{x}, u)$ is a unknown smooth function. It can be seen that affine nonlinear systems are a special case of this class of nonlinear systems.

In the past five years, researchers have proposed different AFC schemes [2, 33-38] for non-affine nonlinear systems. Because the control input does not appear linearly, the well-known feedback linearization technique is not applicable. Adaptive fuzzy control of non-affine nonlinear systems is more difficult and challenging. In general, more advanced mathematical techniques are required.

2.2.2.2. Strict-feedback and pure-feedback nonlinear systems

- Strict-feedback nonlinear systems

A large number of practical nonlinear systems can be expressed in or transformed into a special state-space form called *strict-feedback form*:

$$\begin{cases} \dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1}, & 1 \leq i \leq n-1 \\ \dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)u, & n \geq 2 \\ y = x_1 \end{cases} \quad (2.4)$$

where $\bar{x}_i = [x_1, x_2, \dots, x_i]^T \in R^i$, $i = 1, \dots, n$, $u \in R$, $y \in R$ are state variables, system input and output, respectively. $f_i(\bullet)$ and $g_i(\bullet)$, $i = 1 \dots n$, are smooth unknown functions. The control objective is to determine the control input u such that output y tracks a reference signal r as close as possible.

In the past decade, adaptive backstepping has become one of the most popular design methods for systems in triangular form (2.4) because it can guarantee global stabilities, tracking, and transient performance for the broad class of strict-feedback systems (2.4) with unknown parameters [2]. The idea behind backstepping design is that some appropriate functions of state variables are selected recursively as virtual control inputs for lower dimension subsystems of the overall system. Each backstepping stage results in a new virtual control design, expressed in terms of the virtual control designs from the preceding stages. When the procedure terminates, a feedback design for the true control input results, which achieves the original design objective by virtue of a final Lyapunov functions associated with each individual design stage [39].

However, a major constraint of traditional adaptive backstepping technique is that unknown functions $f_i(\bar{x}_i)$ and $g_i(\bar{x}_i)$, $i=1\dots n$ must be “linear in the unknown parameters”. With the use of neural networks and adaptive fuzzy systems, this assumption can be relaxed.

Adaptive neural network backstepping control has been proposed in [39-42]. Neural networks are used in each step to approximate the unknown functions. A drawback of these adaptive neural network backstepping control schemes is the problem of “explosion of complexity”, the complexity of controllers grows drastically as the order n of the system increases.

This explosion of complexity is caused by the need to estimate derivatives of certain nonlinear functions [43]. At each step, to estimate this derivative, partial derivatives are need to be computed and they are also need to be used as inputs to neural networks. The number of partial derivatives increases drastically after each step, and thus increases drastically the complexity of controllers. To overcome this problem, [43] proposes a dynamic surface control technique, in which a first-order filter is introduced at each step to avoid the need to estimate derivatives of certain nonlinear functions.

Recently, adaptive intelligent control has also been developed for discrete strict-feedback systems. [44] proposes a state-feedback adaptive NN control scheme using backstepping, and an output-feedback adaptive NN control scheme using a diffeomorphism transformation. The MIMO case has also been considered in [45, 46].

- Pure-feedback nonlinear systems

Pure-feedback systems are a broader class of low-triangular-structured nonlinear systems, which is given in a general form as:

$$\begin{cases} \dot{x}_i = f_i(\bar{x}_i, x_{i+1}), & i=1, \dots, n-1 \\ \dot{x}_n = f_n(\bar{x}_n, u) \\ y = x_1 \end{cases} \quad (2.5)$$

where $\bar{x}_i = [x_1, x_2, \dots, x_i]^T \in R^i$, $i=1, \dots, n$, $u \in R$, $y \in R$ are state variables, system input and output, respectively. $f_i(\bar{x}_i, x_{i+1})$, $i=1\dots n$, are smooth functions.

It can be seen that pure-feedback systems (2.5) do not have affine variables as virtual controls, or as the actual control u . Thus, control of pure-feedback systems

(2.5) is more difficult than control of strict-feedback systems (2.4). Few results of controlling pure-feedback systems have been reported in the literature [34, 47].

[47] proposes adaptive neural control of pure-feedback systems by combining backstepping, input-to-state stability analysis, and the small-gain theorem. The proposed control scheme, however, also suffers from the problem of “explosion of complexity”. [34] proposes adaptive neural network control using Nussbaum-Gain functions and the idea of backstepping. A drawback of this approach is the closed-loop system has wild transient performance.

2.2.2.3. SISO and MIMO nonlinear systems

Inspired by the results for SISO nonlinear systems, researchers have also developed adaptive intelligent control for uncertain MIMO nonlinear systems.

Control of uncertain MIMO nonlinear systems, in general, is more difficult. It is due to the difficulties in dealing with the couplings in input matrices and interconnections between subsystems.

[48] proposes adaptive fuzzy control for a class of MIMO nonlinear systems, which consists of affine subsystems. And it is assumed that there is no input coupling and the system interconnections are bounded with known constants.

[49-53] present adaptive fuzzy/neural control for a class of MIMO square nonlinear plants, in which the bounding restrictions on the system interconnections are relaxed. However, it is required that the number of inputs equals the number of outputs and the inputs are also in affine forms.

In [54, 55] adaptive neural network controllers were proposed for some special classes of MIMO nonlinear robotic systems, using several nice properties of the robotic systems.

In [56], an adaptive neural control approach was proposed for a class of MIMO nonlinear systems with a triangular structure in control inputs.

In [57], adaptive neural control is proposed for two classes of uncertain MIMO nonlinear systems in block-triangular forms, which consists of couplings in the inputs as well as in the system interconnections without any bounding restrictions.

Most results available in the literature assume that inputs appear in the affine forms. Control of uncertain MIMO nonlinear systems with nonaffine inputs is still an open problem.

2.2.2.4. State-feedback and output feedback nonlinear systems

State-feedback control deals with systems in which it is assumed that all the state variables are available for measurement. In practice, it is sometime difficult or impossible to measure all the state variables. Output-feedback control is the control of systems in which only outputs are required to be available for measurement.

For affine and nonaffine SISO nonlinear systems, [44, 58] propose adaptive NN output feedback control using high gain observers to estimate the required derivatives of the outputs. Due to the use of high gain observers, a peaking phenomenon in the transient behaviour may occur. To overcome such a problem, saturation methods introduced in [59, 60] may be used. [61, 62] propose using linear observers to observe the error dynamics. [38] proposes a non-observer approach, in which input/output history are used as inputs to NNs instead of the derivatives of the system output.

Adaptive intelligent output feedback control for wider classes of nonlinear systems has also been considered. MIMO cases are considered in [46, 63, 64]. Systems with zero dynamics are treated in [65, 66].

2.2.2.5. Continuous and discrete systems

Since most controllers are implemented using digital computers, control in discrete time domain is an important topic. Adaptive intelligent control for discrete-time nonlinear systems has also received attention from researchers. Due to the difficulties in discrete-time systems, such as the noncausal problem in backstepping design, discrete-time domain methods are much less common than those in the continuous domain [46].

For SISO discrete time systems, [67, 68] propose adaptive intelligent control for a class of discrete affine nonlinear systems. [69] proposes both state and output feedback controls for a class of discrete-time systems with general relative degree and bounded disturbances. For a class of discrete-time systems in strict feedback form, an effective backstepping design method was proposed in [70].

For MIMO discrete time systems, [71] presents adaptive neural network control for affine MIMO nonlinear systems. [45] proposes a state feedback NN control scheme for a class of discrete-time nonlinear MIMO systems with triangular form inputs and bounded disturbances. The authors then present an output feedback control scheme for the same class of MIMO discrete-time systems, in which only input and output sequences are used to construct stable control.

2.2.3. Adaptive mechanism of fuzzy systems

2.2.3.1. Only parameters are tuned

In adaptive intelligent control, intelligent systems (i.e. neural networks, neural-fuzzy systems, or adaptive fuzzy systems) are employed to approximate some unknown functions. To guarantee the stability, parameters of intelligent systems are tuned online.

In an intelligent system, there are two type of parameters: linear parameters and nonlinear parameters. For example, consequents of a fuzzy system are linear parameters, whereas input membership function parameters (centers and variances) are nonlinear parameters. For a multi-layer neural network, synaptic weights of the output layer are linear parameters, whereas weights of the hidden layers are nonlinear parameters.

Most of the work reported in the literature employs intelligent systems with linear tuneable parameters. Fewer results are available for intelligent systems with nonlinear tuneable parameters. [2] proposes adaptive control using multi-layer neural networks, in which the weights of hidden layers are nonlinear parameters. [3, 72, 73] propose adaptive fuzzy control, in which the input membership function parameters are also tuned.

Linear parameterized intelligent systems are simpler to tune and to analyze. They, however, suffer “the curse of dimensionality”, their size tend to increase exponentially with the dimension of the input space. Nonlinear parameterized intelligent systems are normally smaller (in term of size) to achieve the same approximation accuracy and they are global approximators. However, the learning speed is slower and analysis is more difficult. Thus, it normally depends on a particular application to decide which type is more suitable.

2.2.3.2. Both parameters and structure are adjusted

Most intelligent systems used in adaptive control have fixed structures. That is the number of membership functions (in fuzzy systems) or the number of nodes (in neural networks) are fixed. Choosing the right structure is an important aspect as it affects the approximation capability of the intelligent system. It is difficult to choose a suitable structure for a particular application. Normally, a designer needs to try several structures to find a suitable one.

Few attempts to develop self-structuring intelligent systems for adaptive control have been reported. Park et al [37, 74] proposes using self-structuring adaptive fuzzy control, in which rules are added to the rule base as the input space is explored. Gao [51] proposes using self-organising adaptive fuzzy neural control, which is able to add or delete rules from the rule base. Park et al [36] proposes self-structuring adaptive neural network control, in which a neuron in the hidden layer splits into two if a certain condition is satisfied.

However, there exist some limitations in the above methods. Even if self-structuring algorithms are presented, stability analysis is only performed for the fixed-structured case. There is no discussion on the effect of the self-structuring algorithms on the stability. [36, 37, 74] do not propose any algorithm to limit the size of the intelligent systems. Thus, there is a risk that the intelligent systems will exceed the hardware capability if initial performance is poor. Gao [51] uses large matrix manipulation and an Error Reduction Ratio technique to prune rules. Thus, the approach is complicated and computationally inefficient. Self-structuring adaptive intelligent control is, therefore, still an open research topic.

2.3. Preliminaries

2.3.1. Fuzzy system and neural network

The required knowledge includes basic topics such as:

- Fuzzy set theory
- Fuzzy systems (Mamdani and Takagi-Sugeno types)
- Fundamentals of neural networks
- Backpropagation and related training algorithms

There are numerous books in the literature that cover these areas such as [75-77]. Thus, we will not re-present these areas here.

2.3.2. Concepts of stability and boundedness

[2, 3, 78] Consider the autonomous nonlinear system described by

$$\dot{\underline{x}} = \underline{f}(\underline{x}), \underline{x}, \underline{f} \in R^n \quad (2.6)$$

2.3.2.1. Stability definitions

Definition 2.1 A state \underline{x}^* is an equilibrium state (or **equilibrium point**) of the system (2.6), if once $\underline{x}(t)$ is equal to \underline{x}^* , it will remain equal to \underline{x}^* forever. In mathematical terms, that means the vector \underline{x}^* satisfies:

$$f(\underline{x}^*) = 0$$

Without the loss of generality, we may assume the origin $\underline{x}^* = 0$ is an equilibrium point.

Definition 2.2 The equilibrium point $\underline{x}^* = 0$ is said to be **Lyapunov stable** if, for any given $\varepsilon > 0$, there exists a positive $\delta(\varepsilon)$ such that if

$$\|\underline{x}(0)\| < \delta(\varepsilon),$$

$$\text{then } \|\underline{x}(t)\| < \varepsilon, \forall t \geq 0.$$

Otherwise, the equilibrium point is **unstable**.

Definition 2.3 The equilibrium point $\underline{x}^* = 0$ is said to be **asymptotically stable** if it is Lyapunov stable and there exists δ such that if

$$\|\underline{x}(0)\| < \delta,$$

$$\text{then } \lim_{t \rightarrow \infty} \underline{x}(t) = 0.$$

Definition 2.4 The equilibrium point $\underline{x}^* = 0$ is said to be **exponentially stable** if it is asymptotically stable and there exist $\alpha, \beta, \delta > 0$ such that if

$$\|\underline{x}(0)\| < \delta,$$

$$\text{then } \|\underline{x}(t)\| \leq \alpha \|\underline{x}(0)\| e^{-\beta t}, \text{ for } t \geq 0.$$

Conceptually, the meanings of the above terms are the following:

- Lyapunov stability of an equilibrium point means that solutions starting “close enough” to the equilibrium point (within the distance δ from it) remain “close enough” forever. Note that this must be true for any ε that one may want to choose.
- Asymptotic stability means that solutions that start close enough not only remain close enough but also eventually converge to the equilibrium.

- Exponential stability means that solutions not only converge, but in fact converge faster than or at least as fast as a particular known rate $\alpha \|x(0)\| e^{-\beta t}$.

2.3.2.2. Boundedness definitions

Definition 2.5 A solution $\underline{x}(t)$ is **bounded** if there exists a $\beta > 0$, that may depend on each solution, such that

$$\|\underline{x}(t)\| < \beta \text{ for all } t \geq 0.$$

Definition 2.6 The solutions $\underline{x}(t)$ are **uniformly bounded** if for any $\alpha > 0$, there exists $\beta(\alpha)$ such that if

$$\|\underline{x}(0)\| < \alpha,$$

then $\|\underline{x}(t)\| < \beta(\alpha)$ for all $t \geq 0$.

Definition 2.7 The solutions $\underline{x}(t)$ are **uniformly ultimately bounded** if for any $\alpha > 0$, there exist β and $T(\beta, \underline{x}(0))$ such that if

$$\|\underline{x}(0)\| < \alpha,$$

then $\|\underline{x}(t)\| < \beta$ for all $t \geq T(\beta, \underline{x}(0))$.

Definition 2.8 The solutions $\underline{x}(t)$ are **semi-globally uniformly ultimately bounded** if for any Ω , a compact subset of \Re^n , there exist β and $T(\beta, \underline{x}(0))$ such that if

$$\|\underline{x}(0)\| \in \Omega,$$

then $\|\underline{x}(t)\| < \beta$ for all $t \geq T(\beta, \underline{x}(0))$.

2.3.3. Lyapunov stability theorem

Definition 2.9 A continuous function $\gamma : \Re \rightarrow \Re^+$ is said to belong to **class K** if

- $\alpha(0) = 0$.
- $\alpha(r) \rightarrow \infty$ as $r \rightarrow \infty$.
- $\alpha(r) > 0 \quad \forall r > 0$.
- $\alpha(r)$ is non-decreasing, i.e. $\alpha(r_1) \geq \alpha(r_2) \quad \forall r_1 > r_2$.

Definition 2.10 A continuous function $V(x, t) : \Re^m \times \Re^+ \rightarrow \Re$ is

- **locally positive definite** if there exists a class K function $\alpha(\bullet)$ such that

$$V(\underline{x}, t) \geq \alpha(\|\underline{x}\|)$$

for all $t \geq 0$ and $\|\underline{x}\|$ in the neighbourhood N of the origin \mathfrak{R}^n .

- **positive definite** if $N = \mathfrak{R}^n$.
- **(locally) negative definite** if $-V$ is (locally) positive definite.
- **(locally) decreascent** if there exists a class K function $\beta(\bullet)$ such that

$$V(\underline{x}, t) \leq \beta(\|\underline{x}\|)$$

for all $t \geq 0$ and $\|\underline{x}\|$ in (the neighbourhood N of the origin) \mathfrak{R}^n .

2.3.3.1. Conditions for stability

Theorem 2.1 Lyapunov Theorem

Given the non-linear dynamic system

$$\dot{\underline{x}} = \underline{f}(\underline{x}, t), \quad \underline{x}(0) = \underline{x}_0$$

with an equilibrium point at the origin, and let N be a neighbourhood of the origin, i.e. $N = \{\underline{x} : \|\underline{x}\| \leq \varepsilon, \text{ with } \varepsilon > 0\}$, then the origin 0 is

- *stable in the sense of Lyapunov if for $\underline{x} \in N$, there exists a scalar function $V(\underline{x}, t)$ such that $V(\underline{x}, t) > 0$ and $\dot{V}(\underline{x}, t) \leq 0, \forall \underline{x} \neq 0$.*
- *uniformly stable if for $\underline{x} \in N$, there exists a scalar function $V(\underline{x}, t)$ such that $V(\underline{x}, t) > 0$ and decreascent and $\dot{V}(\underline{x}, t) \leq 0, \forall \underline{x} \neq 0$.*
- *asymptotically stable if for $\underline{x} \in N$, there exists a scalar function $V(\underline{x}, t)$ such that $V(\underline{x}, t) > 0$ and $\dot{V}(\underline{x}, t) < 0, \forall \underline{x} \neq 0$.*
- *globally asymptotically stable if for $\underline{x} \in \mathfrak{R}^n$, there exists a scalar function $V(\underline{x}, t)$ such that $V(\underline{x}, t) > 0$ and $\dot{V}(\underline{x}, t) \leq 0, \forall \underline{x} \neq 0$.*
- *uniformly asymptotically stable if for $\underline{x} \in N$, there exists a scalar function $V(\underline{x}, t)$ such that $V(\underline{x}, t) > 0$ and decreascent and $\dot{V}(\underline{x}, t) < 0, \forall \underline{x} \neq 0$.*

- globally, uniformly, asymptotically stable if $\underline{x} \in \mathfrak{R}^n$, there exists a scalar function $V(\underline{x}, t)$ such that $V(\underline{x}, t) > 0$ and decrescent and $\dot{V}(\underline{x}, t) < 0$, $\forall \underline{x} \neq 0$.
- exponentially stable if there exist positive constants α , β , γ such that $\forall \underline{x} \in N$, $\alpha \|\underline{x}\|^2 \leq V(\underline{x}, t) \leq \beta \|\underline{x}\|^2$ and $\dot{V}(\underline{x}, t) \leq -\gamma \|\underline{x}\|^2$.
- globally exponentially stable if there exist positive constants α , β , γ such that $\forall \underline{x} \in \mathfrak{R}^n$, $\alpha \|\underline{x}\|^2 \leq V(\underline{x}, t) \leq \beta \|\underline{x}\|^2$ and $\dot{V}(\underline{x}, t) \leq -\gamma \|\underline{x}\|^2$.

2.3.3.2. Conditions for boundedness

Uniform ultimate boundedness (UUB) If there exists a function $V(\underline{x})$ with continuous partial derivatives such that for $\underline{x} \in S \subset \mathfrak{R}^n$:

- $V(\underline{x})$ is positive definite: $V(\underline{x}) > 0$, $\forall \|\underline{x}\| \neq 0$
- Time derivative of $V(\underline{x})$ is negative definite outside of S:
 $\dot{V}(\underline{x}) < 0$, $\forall \|\underline{x}\| > \beta$, $(\|\underline{x}\| \leq B) \Rightarrow (\underline{x} \in S)$

Then the system is UUB and $\|\underline{x}\| \leq B$, $\forall t \geq t_0 + T$.

2.3.4. Universal approximation properties

2.3.4.1. Universal approximation property for zero-order Takagi-Sugeno fuzzy systems

Consider zero-order Takagi-Sugeno fuzzy systems with point fuzzification method, product-type inference, and center-average defuzzifier.

For each $a < b$, $a, b \in R$, let $\alpha(a, b): R \rightarrow [0, 1]$ be a membership function such that $\alpha(a, b)(x) \neq 0$ if $x \in (a, b)$ and $\alpha(a, b)(x) = 0$ if $x \notin (a, b)$. The fuzzy system has the **If-Then** rule base of the following form:

R⁽ⁱ⁾: IF x_1 is A_1^i , and x_2 is A_2^i , and ...and x_n is A_n^i ,
 THEN y is θ_i

where $x = (x_1, x_2, \dots, x_n)^T \in U \subset R^n$ and $y \in V \subset R$ are the crisp input and output of the fuzzy system. A_j^i are fuzzy sets with membership functions

$A_j^i(x_j) = \alpha(a_{j1}^i, a_{j2}^i)(x_j)$ for some $a_{j1}^i < a_{j2}^i$, $i = 1, \dots, M$ where M is the number of rules, $j = 1, \dots, n$. θ_i is the system output due to rule $R^{(i)}$.

Then, the output of a Takagi-Sugeno fuzzy system is a weighted average of θ_i :

$$\hat{y} = \hat{f}(\underline{x} | \underline{\theta}) = \frac{\sum_{i=1}^M \theta_i \mu_i(\underline{x})}{\sum_{i=1}^M \mu_i(\underline{x})} = \sum_{i=1}^M \theta_i \zeta_i(\underline{x}) \quad (2.7)$$

in which $\mu_i(\underline{x}) = \prod_{j=1}^n A_j^i(x_j)$, $\zeta_i(\underline{x}) \geq 0$ and $\sum_{i=1}^M \zeta_i(\underline{x}) = 1$.

Theorem 2.2: Universal approximation theorem

For any given real continuous function g on a compact set $U \subset \mathbb{R}^n$ and arbitrary $\varepsilon > 0$, if a large enough number of rules is used, there exists a fuzzy logic system f in the form of (2.7) such that

$$\sup_{\underline{x} \in U} |f(\underline{x}) - g(\underline{x})| < \varepsilon$$

Proof

The proof of this theorem can be found in [1, 3].

Remark 2.1 This theorem justifies that Takagi-Sugeno fuzzy systems with either triangular membership functions or Gaussian membership functions are universal approximators. Thus, in this thesis, we will use both Takagi-Sugeno fuzzy systems with triangular membership functions and the ones with Gaussian membership functions as our fuzzy controllers.

Remark 2.2 This theorem is just an existence theorem. How to determine the sufficient number of rules or how to find such a fuzzy logic system are different questions. We are more interested in answer the question “*How to find a fuzzy logic controller such that the closed-loop system is stable and the tracking error converge to a small neighbourhood of zero?*”.

Remark 2.3 The importance of this theorem should not be overemphasized because many other types of functions are also universal approximators (polynomials, neural networks, etc.). What should be emphasized is the capability of the fuzzy logic systems to incorporate linguistic information in a natural and systematic way.

2.4. Basic indirect adaptive fuzzy control for SISO affine nonlinear systems

As an example, this section shows how the above mathematical tools are used to construct a simple adaptive fuzzy controller for SISO affine nonlinear systems. Consider SISO affine nonlinear systems in the following form:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= x_3 \\
 &\dots \\
 &\dots \\
 \dot{x}_n &= f(\underline{x}) + g(\underline{x})u \\
 y &= x_1
 \end{aligned} \tag{2.8}$$

where u is the control input; y is the output; $f(\underline{x})$ and $g(\underline{x})$ are unknown continuous functions; $\underline{x} = (x_1, x_2, \dots, x_n)^T$ is the state vector of the system which is assumed available for measurement.

Control objective is to design an adaptive fuzzy controller such that the output $y(t)$ of the system follows a continuous reference signal $r(t) \in C^n$.

Assumptions

To design a controller satisfying the above control objective, the following assumptions are made:

- **Assumption 2.1:** $g(\underline{x})$ is continuous and the sign of $g(\underline{x})$ is known for $\underline{x} \in \Omega_{\underline{x}}$, where $\Omega_{\underline{x}}$ is the controllability region.

Since $g(\underline{x}) \neq 0$ (controllable condition of system (2.8)) and $g(\underline{x})$ is continuous for \underline{x} in the controllability region $\Omega_{\underline{x}}$, without loss of generality, it can be assumed that $g(\underline{x}) > 0$ for $\underline{x} \in \Omega_{\underline{x}}$.

- **Assumption 2.2:** Define $\underline{r} = [\dot{r}, \ddot{r}, \ddot{\ddot{r}}, \dots, r^{(n-1)}]^T$. We assume that $\|\underline{r}\| \leq r_0$ and $\|r^{(n)}\| \leq r_1$ with known constants $r_0, r_1 > 0$.

Ideal control

Let $e = r - y$, $\underline{e} = (e, \dot{e}, \ddot{e}, \dots, e^{(n-1)})^T$, and $\underline{k} = (k_1, k_2, \dots, k_n)^T$ be such that the polynomial $s^n + k_n s^{n-1} + \dots + k_1$ is Hurwitz stable. If the functions $f(\underline{x})$ and $g(\underline{x})$ are known, then the control law

$$u^* = \frac{1}{g(\underline{x})} \left(-f(\underline{x}) + \underline{k}^T \underline{e} + r^{(n)} \right) \tag{2.9}$$

applied to (2.8) results in

$$\dot{e}^{(n)} = -\underline{k}^T \underline{e} = -k_1 e - k_2 \dot{e} \dots - k_n e^{(n-1)} \quad (2.10)$$

which implies that $\lim_{t \rightarrow +\infty} e = 0$. The control u^* is called *ideal control*.

Certainty equivalent control, direct and indirect AFC

However, $f(\underline{x})$ and $g(\underline{x})$ are unknown. Thus, we need to employ fuzzy systems to approximate the unknown functions. If we use one fuzzy system to approximate u^* , we have *direct AFC*. If we use two fuzzy systems to model $f(\underline{x})$ and $g(\underline{x})$, we have *indirect AFC*. Direct AFC will be discussed in the next chapter. Here, we consider the indirect case.

Employ two fuzzy systems $\hat{f}(\underline{x}|\underline{\theta}_f)$ and $\hat{g}(\underline{x}|\underline{\theta}_g)$ in the form (2.7) to approximate $f(\underline{x})$ and $g(\underline{x})$ respectively. The resulting control law is

$$u_c = \frac{1}{\hat{g}(\underline{x}|\underline{\theta}_g)} \left(-\hat{f}(\underline{x}|\underline{\theta}_f) + \underline{k}^T \underline{e} + r^{(n)} \right) \quad (2.11)$$

is the so-called *certainty equivalent control*.

Ideal parameters and minimum approximation error

The *ideal parameters* $\underline{\theta}_f^*$ and $\underline{\theta}_g^*$ are defined as:

$$\underline{\theta}_f^* = \arg \min \left[\sup_{\underline{x} \in U_x} \left| \hat{f}(\underline{x}|\underline{\theta}_f) - f(\underline{x}) \right| \right] \quad (2.12)$$

$$\underline{\theta}_g^* = \arg \min \left[\sup_{\underline{x} \in U_x} \left| \hat{g}(\underline{x}|\underline{\theta}_g) - g(\underline{x}) \right| \right] \quad (2.13)$$

The minimum approximation error is defined as:

$$\omega_f = \left(\hat{f}(\underline{x}|\underline{\theta}_f^*) - f(\underline{x}) \right) \quad (2.14a)$$

$$\omega_g = \left(\hat{g}(\underline{x}|\underline{\theta}_g^*) - g(\underline{x}) \right) \quad (2.14b)$$

Stability analysis and adaptive laws

Substituting $u = u_c$, adding and subtracting $g(\underline{x})u^*$ to (2.8), we obtain the error equation

$$\dot{e}^{(n)} = -\underline{k}^T \underline{e} + \left(\hat{f}(\underline{x}|\underline{\theta}_f) - f(\underline{x}) \right) + \left(\hat{g}(\underline{x}|\underline{\theta}_g) - g(\underline{x}) \right) u_c \quad (2.15)$$

or in the matrix form

$$\dot{\underline{e}} = \Lambda_c \underline{e} + \underline{b}_c \left[\left(\hat{f}(\underline{x}|\underline{\theta}_f) - f(\underline{x}) \right) + \left(\hat{g}(\underline{x}|\underline{\theta}_g) - g(\underline{x}) \right) u_c \right] \quad (2.16)$$

where

$$\Lambda_C = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -k_1 & -k_2 & -k_3 & \cdots & -k_n \end{pmatrix}, \underline{b}_C = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}.$$

From (2.14), (2.16) becomes

$$\dot{\underline{e}} = \Lambda_C \underline{e} + \underline{b}_C \left[\left(\hat{f}(\underline{x} | \underline{\theta}_f) - \hat{f}(\underline{x} | \underline{\theta}_f^*) \right) + \left(\hat{g}(\underline{x} | \underline{\theta}_g) - \hat{g}(\underline{x} | \underline{\theta}_g^*) \right) \underline{u}_c \right] + \underline{b}_C \omega \quad (2.17)$$

where the total approximation error $\omega = \omega_f + \omega_g \underline{u}_c$.

From (2.7), (2.17) can be written as

$$\dot{\underline{e}} = \Lambda_C \underline{e} + \underline{b}_C \left[\underline{\phi}_f^T \underline{\zeta}(\underline{x}) + \underline{\phi}_g^T \underline{\zeta}(\underline{x}) \underline{u}_c \right] + \underline{b}_C \omega \quad (2.18)$$

where $\underline{\phi}_f = \underline{\theta}_f - \underline{\theta}_f^*$, $\underline{\phi}_g = \underline{\theta}_g - \underline{\theta}_g^*$.

Since Λ_C is a stable matrix, there exists a unique positive definite symmetric $n \times n$ matrix P which satisfies the Lyapunov equation:

$$\Lambda_C^T P + P \Lambda_C = -Q \quad (2.19)$$

where Q is an arbitrary $n \times n$ positive definite matrix.

To perform *the stability analysis*, consider the Lyapunov function candidate

$$V = \frac{1}{2} \underline{e}^T P \underline{e} + \frac{1}{2\gamma_f} \underline{\phi}_f^T \underline{\phi}_f + \frac{1}{2\gamma_g} \underline{\phi}_g^T \underline{\phi}_g \quad (2.20)$$

where γ_f and γ_g are positive constants. The time derivative of V along the trajectory of (2.18) is

$$\begin{aligned} \dot{V} = & -\frac{1}{2} \underline{e}^T Q \underline{e} + \underline{e}^T P \underline{b}_C \omega \\ & + \frac{1}{\gamma_f} \underline{\phi}_f^T \left[\dot{\underline{\theta}}_f + \gamma_f \underline{e}^T P \underline{b}_C \underline{\zeta}(\underline{x}) \right] + \frac{1}{\gamma_g} \underline{\phi}_g^T \left[\dot{\underline{\theta}}_g + \gamma_g \underline{e}^T P \underline{b}_C \underline{\zeta}(\underline{x}) \right] \end{aligned} \quad (2.21)$$

where we used (2.19) and $\dot{\underline{\phi}}_f = \dot{\underline{\theta}}_f$, $\dot{\underline{\phi}}_g = \dot{\underline{\theta}}_g$. If we choose *the adaptive laws*

$$\dot{\underline{\theta}}_f = -\gamma_f \underline{e}^T P \underline{b}_C \quad (2.22)$$

$$\dot{\underline{\theta}}_g = -\gamma_g \underline{e}^T P \underline{b}_C \quad (2.23)$$

then from (2.21) we have

$$\dot{V} = -\frac{1}{2} \underline{e}^T Q \underline{e} + \underline{e}^T P \underline{b}_C \omega \quad (2.24)$$

From the universal approximation theorem (theorem 2.2), if a sufficient number of rules is selected, ω should be small if not equal to zero. If $\omega = 0$, (2.24) becomes

$$\dot{V} = -\frac{1}{2} \underline{e}^T Q \underline{e} \leq 0. \quad (2.25)$$

Since V is lower bounded (≥ 0) and \dot{V} is uniformly continuous, using the Barbalat's Lemma (lemma 2.1), we have $\lim_{t \rightarrow +\infty} \dot{V} = 0$, therefore $\lim_{t \rightarrow \infty} |\underline{e}(t)| = 0$.

This completes the design of the basic indirect AFC of affine nonlinear systems. It has been shown that, for system (2.8), if the controller is chosen as (2.11), adaptive laws (2.22), (2.23), and *sufficient number of rules* for fuzzy systems $\hat{f}(\underline{x}|\underline{\theta}_f)$ and $\hat{g}(\underline{x}|\underline{\theta}_g)$ are selected, then the system output will follow the reference signal.

In summary, the design procedure of an AFC system consists of the following steps:

- Show the existence of an ideal control: assume all functions are known, show that there exists a control such that the control objectives are met.
- Show that there exist fuzzy systems to approximate the unknown functions.
- Define the ideal parameters and approximation errors
- Choose a suitable Lyapunov function to derive adaptive laws such that the control objectives are met.

The presented controller is one of the simplest forms of AFC, which was proposed in the early 1990s [1]. Some of its main limitations are discussed in the remarks bellows.

Remark 2.4 The above analysis assumes that the approximation error is small and can be neglected. It is often in practice that the approximation error cannot be ignored. Thus, extra efforts are normally needed to account for the approximation error. In [1, 5, 13, 30, 79], the analysis of stability is only valid under the assumption that the approximation error is square integrable. Some researchers suggested an addition of a variable structure control term to the control law [3, 17, 18, 62, 80]. A number of researchers propose some approaches to estimate the upper bound of the approximation errors [4, 12, 33, 81, 82].

Remark 2.5 Adaptive laws (2.22), (2.23), do not guarantee the boundedness of the fuzzy parameters. To overcome this problem, modified adaptive laws such as projection algorithms [1, 17], ε -modified and σ -modified adaptive laws [3, 55] have been proposed in the literature.

Remark 2.6 The singularity problem may occur, i.e. the control (2.11) is indefinite if fuzzy system $\hat{g}(\underline{x}|\underline{\theta}_g)$ approaches zero. In practice, extra attentions are needed to prevent this. Ge et al [2, 55] assumes $\frac{\partial g(\underline{x})}{\partial x_n} = 0$ to design novel adaptive controllers while avoiding the singularity problem. Chen and Liu [83] suggest that the initial values of the NN weights need to be chosen sufficiently close to the ideal values. Thus, offline pre-training is needed. Other methods include using projection algorithms [1, 3, 17], a smooth projection algorithm [17], and introducing switching control portions to keep the control magnitudes bounded [55].

Remark 2.7 Even it is shown that the state vector converges to zero, the state vector \underline{x} is not guaranteed to stay in the desired set $U_{\underline{x}}$. To keep the state variables of the system under control in a desired constraint set without the need of large adaptive gains, some researchers [1, 13, 31, 32] propose adaptive fuzzy control combined with a supervisory control. When the state variables are well inside the constraint set, the supervisory control is zero. When the state variables tend to move outside of the desired boundaries, the supervisory control begins to operate to force the states stay in the constraint set.

2.5. Conclusion

A general review of AFC has been given in this chapter. The review has shown the rapid development of AFC in the past decade, which results in the diversity and variety of AFC schemes available in the literature. It also shown that there are still limitations and areas that need to improve.

Stability concepts and Lyapunov stability techniques are the main mathematical tools that are use throughout the thesis. These mathematical tools have also been presented in section 1.3.

Finally, the basic framework of an AFC scheme has been presented through an indirect AFC of affine nonlinear systems. Basic concepts such as ideal control, ideal parameters, minimum approximation error, and adaptive laws have been introduced.

A main drawback of the presented indirect AFC is the effect of the approximation error. In the next chapter, we will present how to compensate this by utilising an approximation error estimator and an automatic switching mechanism.

3. Chapter 3

TWO-MODE INDIRECT ADAPTIVE FUZZY CONTROL WITH APPROXIMATION ERROR ESTIMATOR¹

3.1. Introduction

One limitation of the indirect AFC scheme presented in section 2.4 is the effect of the approximation error. In this chapter, a two-mode indirect adaptive fuzzy control with approximation error estimator is proposed. Equipped with a switching mechanism, the controller is also able to automatically switch between two modes, learning mode and operating mode, to reduce the number of parameters needed to be tuned online.

In section 3.2, a short survey about the effect of the approximation error in AFC is given first. Then, the two-mode indirect AFC scheme is presented in section 3.3. Section 3.4 shows application to an inverted pendulum and a Chua's chaotic circuit to demonstrate the proposed control scheme. Finally, some conclusions are drawn in section 3.5.

For the continuity of reading, all figures are displayed at the end of the chapter.

3.2. Literature review

The result in section 2.4 assumes that the approximation error is small and can be neglected. It is often the case that in practice the approximation error cannot be ignored. In [1, 5, 13, 30, 79], the analysis of stability is only valid under the assumption that the approximation error is square integrable. Some researchers suggest adding a variable structure control term to the control law [3, 17, 18, 62, 80]. Other researchers [2, 12, 33, 81, 82] propose to estimate the upper bound of the approximation errors.

Park [81] solves this problem by estimating these bounds using fuzzy inference. This requires manual tuning of fuzzy estimators. Er [12] propose using a non-negative adaptive law to update the estimators. Thus, the estimated bounds are unbounded. Sun et al [82], Park [33], and Ge [2] also present solutions in which they propose using a

¹ The content of this chapter has been published in IEEE Transactions on Fuzzy Systems :

P.A. Phan, and T.J. Gale, "Two-mode adaptive fuzzy control with approximation error estimator", *IEEE Transactions on Fuzzy Systems*, volume 15 (5), pp 943-955, Oct. 2007.

σ -modification adaptive law to update the estimators. This guarantees the boundedness of the estimated bounds.

In [42, 84, 85] bound estimators are also proposed but it is assumed that fuzzy models of the plants are already available. No algorithm to tune fuzzy system parameters is provided. Designers need to design fuzzy systems manually. The advantage of these controllers is that they need only a few adaptive parameters regardless of the complexity of the controlled plant, and thus, they are more computationally efficient.

To distinguish the above two cases, we refer to a controller as being in **learning mode** when its fuzzy parameters are tuned online and as being in **operating mode** when its fuzzy parameters are fixed.

One may wonder whether it is possible to design an adaptive fuzzy controller that can operate in the aforementioned modes, learning mode and operating mode. Obviously, this controller would be better since it has the advantages of both modes: learning ability and computational efficiency. And if the answer is yes, how can one decide which mode the controller should be in? This motivates us to propose a 2-mode adaptive fuzzy controller with approximation error estimator.

3.3. Two-mode adaptive fuzzy control with approximation error estimator

We consider the same control problem as in section 2.4. The nonlinear system is given as:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= x_3 \\
 &\dots \\
 &\dots \\
 \dot{x}_n &= f(\underline{x}) + g(\underline{x})u \\
 y &= x_1
 \end{aligned} \tag{3.1}$$

with the control objective and two assumptions given in section 1.4. The following two additional assumptions are also required:

- **Assumption 3.3:** We can determine parameter vectors $\underline{\theta}_f^L$, $\underline{\theta}_f^U$, $\underline{\theta}_g^L$ and $\underline{\theta}_g^U$ such that $\hat{f}(\underline{x}|\underline{\theta}_f^U) \geq f(\underline{x}) \geq \hat{f}(\underline{x}|\underline{\theta}_f^L)$ and $\hat{g}(\underline{x}|\underline{\theta}_g^U) \geq f(\underline{x}) \geq \hat{g}(\underline{x}|\underline{\theta}_g^L)$, $\forall \underline{x} \in \Omega_{\underline{x}}$.

- **Assumption 3.4:** The considered plant is slow time-varying and there are small disturbances so that, during the operation time, there exist $\underline{\theta}_f^0$, $\underline{\theta}_g^0$ such that:

$$\omega^0 = \left| f(\underline{x}) - \hat{f}(\underline{x}|\underline{\theta}_f^0) + (g(\underline{x}) - \hat{g}(\underline{x}|\underline{\theta}_g^0))\mu \right| \leq \omega_{\max}, \quad \forall \underline{x} \in \Omega_{\underline{x}}.$$

We employ zero-order Takagi-Sugeno fuzzy systems with Gaussian membership functions for input, center-average defuzzifier, and product-type inference. From (2.7), the output of a fuzzy system of this type is:

$$\hat{y} = \hat{f}(\underline{x}|\underline{\theta}) = \frac{\sum_{i=1}^M \theta_i \mu_i(\underline{x})}{\sum_{i=1}^M \mu_i(\underline{x})} = \sum_{i=1}^M \theta_i \zeta_i(\underline{x})$$

in which $\mu_i(\underline{x})$ are product of Gaussian membership functions ,i.e.

$$\mu_i(\underline{x}) = \prod_{j=1}^n \exp\left(-\frac{1}{2}\left(\frac{x_j - c_j^i}{\sigma_j^i}\right)^2\right), \quad \text{where } c_j^i \text{ and } \sigma_j^i \text{ are the centers and widths of the}$$

membership functions for the j^{th} input and the i^{th} rule.

The two-mode indirect adaptive fuzzy control is proposed as follows.

The control signal

$$\begin{aligned} u &= u_{ce} + u_s \\ &= \frac{1}{\hat{g}(\underline{x}|\underline{\theta}_g)} \left(-\hat{f}(\underline{x}|\underline{\theta}_f) + \underline{k}^T \underline{e} + r^{(n)} \right) + \frac{1}{\hat{g}(\underline{x}|\underline{\theta}_g)} \hat{\omega} \tanh\left(\frac{0.2785 \underline{e}^T P \underline{b}_c \hat{\omega}}{\varepsilon}\right) \end{aligned} \quad (3.2)$$

where ε is a small constant specified by designers, $\hat{\omega}$ is the variable used to estimate the approximation error, and other parameters are defined as in section II.

The adaptive laws

$$\dot{\theta}_{fi} = \begin{cases} -\gamma_f \underline{e}^T P \underline{b}_c \zeta_{fi}(\underline{x}) & \text{if } (\theta_{fi}^L < \theta_{fi} < \theta_{fi}^U) \\ & \text{or } (\theta_{fi} = \theta_{fi}^L \text{ and } \gamma_f \underline{e}^T P \underline{b}_c \zeta_{fi}(\underline{x}) \leq 0) \\ & \text{or } (\theta_{fi} = \theta_{fi}^U \text{ and } \gamma_f \underline{e}^T P \underline{b}_c \zeta_{fi}(\underline{x}) \geq 0) \\ 0 & \text{if } (\theta_{fi} = \theta_{fi}^L \text{ and } \gamma_f \underline{e}^T P \underline{b}_c \zeta_{fi}(\underline{x}) > 0) \\ & \text{or } (\theta_{fi} = \theta_{fi}^U \text{ and } \gamma_f \underline{e}^T P \underline{b}_c \zeta_{fi}(\underline{x}) < 0) \end{cases} \quad (3.3)$$

in which γ_f is the adaptive gain of the fuzzy system $\hat{f}(\underline{x}|\underline{\theta}_f)$, \underline{c}_i is the center of i^{th}

rule, $i = 1, \dots, M_f$, M_f is the number of rules of $\hat{f}(\underline{x}|\underline{\theta}_f)$.

$$\dot{\theta}_{gj} = \begin{cases} -\gamma_g \underline{e}^T P b_C \zeta_{gj}(\underline{x}) & \text{if } (\theta_{gj}^L < \theta_{gj} < \theta_{gj}^U) \\ \quad \text{or } (\theta_{gj} = \theta_{gj}^L \text{ and } \gamma_g \underline{e}^T P b_C \zeta_{gj}(\underline{x}) \leq 0) \\ \quad \text{or } (\theta_{gj} = \theta_{gj}^U \text{ and } \gamma_g \underline{e}^T P b_C \zeta_{gj}(\underline{x}) \geq 0) \\ 0 & \text{if } (\theta_{gj} = \theta_{gj}^L \text{ and } \gamma_g \underline{e}^T P b_C \zeta_{gj}(\underline{x}) > 0) \\ \quad \text{or } (\theta_{gj} = \theta_{gj}^U \text{ and } \gamma_g \underline{e}^T P b_C \zeta_{gj}(\underline{x}) < 0) \end{cases} \quad (3.4)$$

in which γ_g is the adaptive gain of the fuzzy system $\hat{g}(\underline{x}|\underline{\theta}_g)$, \underline{c}_j is the center of j^{th}

rule, $j = 1, \dots, M_g$, M_g is the number of rules of $\hat{g}(\underline{x}|\underline{\theta}_g)$

$$\dot{\hat{\omega}} = \begin{cases} \gamma_\omega \left(\underline{e}^T P b_C - \sigma_\omega (\hat{\omega} - \omega_0) \right) & \text{if } (\hat{\omega} < \omega_{\max}) \\ \quad \text{or } (\hat{\omega} = \omega_{\max} \text{ and } |\underline{e}^T P b_C| \leq \sigma_\omega (\hat{\omega} - \omega_0)) \\ 0 & \text{if } (\hat{\omega} = \omega_{\max} \text{ and } |\underline{e}^T P b_C| > \sigma_\omega (\hat{\omega} - \omega_0)) \end{cases} \quad (3.5)$$

in which γ_ω is the adaptive gain of the estimator $\hat{\omega}$. And σ_ω , ω_0 , and $\omega_{\max} (\geq W)$ are design parameters specified by designers.

The stability of the controller is stated in two theorems below.

Theorem 3.1 Stability in the learning mode

Consider the system (3.1). If assumptions 3.1-3.3 are satisfied, then an adaptive fuzzy controller with control signal (3.2) and the adaptive laws (3.3), (3.4), (3.5) guarantees that:

(a) The closed-loop system is stable in the sense that all the variables are bounded. In particular,

$$i) \theta_{fi}^L \leq \theta_{fi} \leq \theta_{fi}^U, \quad i = 1 \dots M_f \text{ and } 0 < \theta_{gj}^L \leq \theta_{gj} \leq \theta_{gj}^U, \quad j = 1 \dots M_g.$$

$$ii) \omega_0 \leq \hat{\omega} \leq \omega_{\max}.$$

$$iii) |\underline{e}| \leq \sqrt{\frac{2 \max(V(0), V_r)}{\lambda_{\min}(P)}}, \text{ where } \lambda_{\min}(P) \text{ is the minimum eigen value of}$$

P . $V(0)$ and V_r are bounded positive constants.

$$iv) |\underline{x}| \leq |\underline{r}| + |\underline{e}| \leq r_0 + \sqrt{\frac{2 \max(V(0), V_r)}{\lambda_{\min}(P)}}.$$

v) The bound of $|\underline{u}|$ is

$$\frac{1}{\hat{g}(\underline{x}|\underline{\theta}_g^L)} \left(\max \left(\left| \hat{f}(\underline{x}|\underline{\theta}_f^L) \right|, \left| \hat{f}(\underline{x}|\underline{\theta}_f^U) \right| \right) + r_1 + |\underline{k}| \sqrt{\frac{2 \max(V(0), V_r)}{\lambda_{\min}(P)}} + \omega_{\max} \right), \underline{x} \in \Omega_{\underline{x}}.$$

(b) The tracking error converges to a small neighborhood D_e of zero:

$$D_e = \left\{ |e| : |e| \leq \sqrt{\frac{2V_r}{\lambda_{\min}(P)}} \right\}.$$

(c) The Root Mean Square (RMS) of the tracking error is bounded by

$$RMS = \sqrt{\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t e^2 dt} \leq \sqrt{\frac{2d}{\lambda_{\min}(Q)}}$$

where d is a bounded positive constant.

Proof: the proof is given in appendix 3.A◇

Theorem 3.2 Stability in the operating mode

Consider the system (1). If assumptions 3.1-3.4 are satisfied, then an adaptive fuzzy controller with control signal (3.2) and the adaptive law (3.5) guarantees that:

(a) The closed-loop system is stable:

$$i) \quad \omega_0 \leq \hat{\omega} \leq \omega_{\max}.$$

$$ii) \quad |\underline{e}| \leq \sqrt{\frac{2 \max(V_1(0), V_{1r})}{\lambda_{\min}(P)}}.$$

in which $V_1(0)$ and V_{1r} are bounded positive constants.

$$iii) \quad |\underline{x}| \leq r_0 + \sqrt{\frac{2 \max(V_1(0), V_{1r})}{\lambda_{\min}(P)}}.$$

iv) The bound of $|u|$ is

$$\frac{1}{\hat{g}(\underline{x}|\underline{\theta}_g^0)} \left(\left| \hat{f}(\underline{x}|\underline{\theta}_f^0) \right| + r_1 + |\underline{k}| \sqrt{\frac{2 \max(V_1(0), V_{1r})}{\lambda_{\min}(P)}} + \omega_{\max} \right), \underline{x} \in U_{\underline{x}}.$$

(b) The tracking error converges to a small neighbourhood D'_e of zero

$$D'_e = \left\{ |e| : |e| \leq \sqrt{\frac{2V_{1r}}{\lambda_{\min}(P)}} \right\}.$$

(c) The RMS error is bounded by:

$$RMS \leq \sqrt{\frac{2d}{\lambda_{\min}(Q)}}.$$

Proof: the proof is given in appendix 3.B◇

Switching mechanism The switching between two modes is performed automatically by the following mechanism

- Step 1: parameter initialising.

Using available linguistic knowledge, we construct initial fuzzy

systems $\hat{f}(x|\underline{\theta}_f(0))$ and $\hat{g}(x|\underline{\theta}_g(0))$.

- Step 2: learning mode

Use the controller described by theorem 1.

Switch to the learning phase when $\underline{e}^T P \underline{b}_C$ is smaller than a pre-defined value

E_0 for a specified time interval ΔT_0 .

- Step 3: operating mode

Turn off the parameter update algorithm. Use the controller described by theorem 2 with only one adaptive parameter, which is the estimator value $\hat{\omega}$.

Go back to step 2 if $\underline{e}^T P \underline{b}_C$ is larger than E_0 .

The flow chart is given in figure 3.1.

Remark 3.1 Theorems 3.1 and 3.2 show that the performance of the controller depends on positive constants $V(0)$, V_r , d , $V_1(0)$, and V_{lr} . Even though we cannot determine these values exclusively, their definitions (defined in appendix 3.A and 3.B) suggest that we can make them arbitrarily small by tuning appropriate parameters. Therefore, desired performance can be achieved by these parameters. The intuitive ways to tune the controller are summarized in table 3.1. Often, the choice of which parameters to adjust is dictated by the control problem.

Remark 3.2 An advantage of the 2-mode controller is the reduction of implementation cost. In the learning mode, if the fuzzy system has s inputs and at most two membership functions overlap in each input dimension, there are 2^s adaptive parameters needed to be tuned online. Whereas, in the operating mode, the controller requires only one adaptive parameter no matter what the number of inputs s is. This computational advantage becomes apparent if the controlled plant is a high-order system, in which fuzzy systems with large numbers of inputs are required to represent it.

Remark 3.3 So far, the affect of noise has not been included for clarity. If we consider system (3.1) with bounded noise:

Action	Parameters tuned
To reduce the bound on the error vector $ \underline{e} $	$\uparrow \lambda_{\min}(P), \gamma_f, \gamma_g, \gamma_\omega, \lambda_{\min}(Q)$ $\downarrow \sigma_\omega, \varepsilon, W, \lambda_{\max}(P), V(0)$
To reduce the bound on the state vector $ \underline{x} $	$\downarrow r_0, \underline{e} $
To reduce the bound on the control signal $ \underline{u} $	$\downarrow \max(r^{(n)}, \underline{k} , \underline{e})$
To reduce the bound on $\lim_{t \rightarrow +\infty} \underline{e} $	Similar to how to reduce the bound on $ \underline{e} $, except to reduce $V(0)$
To reduce the RMS error	$\uparrow \lambda_{\min}(Q)$ $\downarrow \sigma_\omega, \varepsilon, W$

Table 3.1: methods to tune the controller's parameters

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
&\dots \\
&\dots \\
\dot{x}_n &= f(\underline{x}) + g(\underline{x})u + d(t) \\
y &= x_1
\end{aligned} \tag{3.6}$$

in which $|d(t)| \leq D$, D is a bounded positive constant, it can be seen that $d(t)$ can be considered as a part of the approximation error when approximating $f(\underline{x})$. Thus, the analysis can be performed as above.

Remark 3.4 One limitation of our proposed 2-mode adaptive fuzzy controller is that it does not have the ability to automatically adjust its structure. In further chapters, we will develop self-structured AFC that are able to automatically adjust their structure.

3.4. Applications

To demonstrate how the proposed 2-mode controller can reduce the number of adaptive parameters, its applications to an inverted pendulum and a Chua's chaotic circuit are presented.

3.4.1. Control of an inverted pendulum

The controlled variable is the angular position of the pendulum (Fig 3.2). The control input is the force applied on the cart. The dynamics of the system is given by:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \left[g \cdot \sin x_1 - \frac{m \cdot l \cdot x_2^2 \cdot \cos x_1 \cdot \sin x_1}{m_c + m} \right] \cdot \frac{1}{l \left(\frac{4}{3} - \frac{m \cos^2 x_1}{m_c + m} \right)} \\
&\quad + \frac{\cos x_1}{m_c + m} \cdot \frac{1}{l \left(\frac{4}{3} - \frac{m \cos^2 x_1}{m_c + m} \right)} \cdot u \\
&= f(\underline{x}) + g(\underline{x})u \\
y &= x_1
\end{aligned} \tag{3.7}$$

in which x_1 is the angular position of the pendulum. x_2 is the angular velocity of the pendulum. m_c is mass of the cart. m is mass of the pendulum and l is half-length of the pendulum. For simulation purpose, $m_c = 1\text{kg}$, $m = 0.1\text{kg}$, and $l = 0.5\text{m}$.

The initial state is $[x_1(0), x_2(0)]^T = [-\pi/60, -\pi/60]^T$.

The control objective is to make the output $y = x_1$ track the reference signal $r(t) = 0.5 \sin(t)$.

Now, we construct the controller as follow

- step 1: let $\Omega_{\underline{x}} = \{(x_1, x_2) \mid |x_1| \leq 1, |x_2| \leq 1\}$
- step 2: construct $\hat{f}(\underline{x} \mid \underline{\theta}_f)$.

Define 5 fuzzy sets each for x_1 , x_2 as shown in Fig 3.3. We assumed that all the possible rules were used. Thus, there are $5 \times 5 = 25$ rules. Examining $f(\underline{x})$, we observe that $-10 < f(\underline{x}) < 10, \forall \underline{x} \in \Omega_{\underline{x}}$. Thus, it is safe to set $\hat{f}(\underline{x} \mid \underline{\theta}_f^U) = 10$ and $\hat{f}(\underline{x} \mid \underline{\theta}_f^L) = -10, \forall \underline{x} \in \Omega_{\underline{x}}$. Then, all the consequences θ_{fi} were initially chosen as $\theta_{fi}(0) = 0, i = 1 \dots 25$.

- step 3: construct $\hat{g}(\underline{x} \mid \underline{\theta}_g)$.

Use the same fuzzy sets for x_1 and x_2 as used in $\hat{f}(\underline{x} \mid \underline{\theta}_f)$. Examining $g(\underline{x})$, we note that $1 \leq g(\underline{x}) \leq 2, \forall \underline{x} \in \Omega_{\underline{x}}$. Therefore, we set $\hat{g}(\underline{x} \mid \underline{\theta}_g^U) = 2$ and $\hat{g}(\underline{x} \mid \underline{\theta}_g^L) = 1$. All the consequences θ_{gi} are chosen as $\theta_{gi}(0) = 1, i = 1 \dots 25$.

- step 4: choose the controller's parameters.

The controller's parameters are chosen as follow:

$$\underline{k} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad Q = \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} \quad P = \begin{pmatrix} 15 & 5 \\ 5 & 5 \end{pmatrix}$$

$$\gamma_f = (f^U - f_L) = 30 \quad \gamma_g = (g^U - g_L) = 1.5$$

$$\gamma_\omega = 0.5 \quad \sigma_\omega = 0.05 \quad \omega_0 = 0.1 \quad \omega_{\max} = 1 \quad \varepsilon = 0.01$$

All the parameters are chosen by using the methods given in table 3.1 in order to achieve the desired performance.

- step 5: design the switching mechanism.

To design the switching mechanism, we need to choose E_0 and ΔT_0 . In this application, we choose $E_0 = 0.1$ and $\Delta T_0 = 10s$.

The simulation results are shown in Fig 3.4. It can be seen that the controller successfully controls the angular position of the inverted pendulum. After about 20s, the tracking error is smaller than 0.01rad. Fig 3.4c shows that the state vector \underline{x} stays in the control region $\Omega_{\underline{x}}$ for all time. Fig 3.4d shows that the control output is quite smooth and there is no chattering. Fig 3.4f shows the control mode, 1 indicates learning mode and 0 indicates operating mode. From Fig 3.4f, we observe that the controller switches from learning mode to operating mode at around 26.8s. Thus, the number of adaptive parameters reduces from 50 (learning mode) to 1 (operating mode). In some cases, the controller may switch between the two modes a few times before actually stay in the operating mode. Whatever the mode the controller is in, the stability is always guaranteed.

Fig 3.4e shows the value of the estimator and demonstrates its typical behaviour. From the start of the simulations, the estimated bounds increase quickly (and is bounded by ω_{\max}) to compensate for the large approximation errors. Later on, when the errors are smaller, the estimated bounds decreases so that no unnecessary excessive control occurs.

3.4.2. Control of a Chua's chaotic circuit

A typical Chua's chaotic circuit consists of one linear resistor, two capacitors, one inductor, and one piecewise-linear resistor. And the original dynamic equations of a Chua's are not in the standard canonical form. However, using a linear transformation, we can transform the dynamic equations into the canonical form. For

simplicity, here we only show the transformed dynamic equations of a Chua's system described in [13] as follow

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= f(\underline{x}) + g(\underline{x})u \\ y &= x_1\end{aligned}\tag{3.8}$$

in which

$$f(\underline{x}) = \frac{14}{1805}x_1 - \frac{168}{9025}x_2 + \frac{1}{38}x_3 - \frac{2}{45}\left(\frac{28}{361}x_1 + \frac{7}{95}x_2 + x_3\right)^3 \text{ and } g(\underline{x}) = 1. \text{ The}$$

initial states are chosen randomly as $x_1(0) = -0.8$, $x_2(0) = 0.2$, $x_3(0) = 0.9$.

The control objective is to control the state x_1 to follow the reference desired signal $r(t) = 1.5 \sin(t)$.

We construct the controller as follow

- step 1: let $\Omega_{\underline{x}} = \{(x_1, x_2, x_3) \mid |x_1| \leq 1, |x_2| \leq 1, |x_3| \leq 1\}$
- step 2: construct $\hat{f}(\underline{x}|\underline{\theta}_f)$.

Define 3 fuzzy sets each for x_1 , x_2 , and x_3 as shown in Fig 3.5. We assumed that all the possible rules were used. Thus, there are $3 \times 3 \times 3 = 27$ rules. Examining $f(\underline{x})$, we observe that $-2 < f(\underline{x}) < 2, \forall \underline{x} \in \Omega_{\underline{x}}$. Thus, it is safe to set $\hat{f}(\underline{x}|\underline{\theta}_f^U) = 2$ and $\hat{f}(\underline{x}|\underline{\theta}_f^L) = -2, \forall \underline{x} \in \Omega_{\underline{x}}$. Then, all the consequences θ_{fi} were initially chosen as $\theta_{fi}(0) = 0, i = 1 \dots 27$.

- step 3: construct $\hat{g}(\underline{x}|\underline{\theta}_g)$.

Use the same fuzzy sets for x_1 , x_2 , and x_3 as used in $\hat{f}(\underline{x}|\underline{\theta}_f)$. We note that $g(\underline{x}) = 1, \forall \underline{x} \in \Omega_{\underline{x}}$, Therefore, we can set $\hat{g}(\underline{x}|\underline{\theta}_g^U) = 1.1$ and $\hat{g}(\underline{x}|\underline{\theta}_g^L) = 0.9$. All the consequences θ_{gi} are chosen as $\theta_{gi}(0) = 1, i = 1 \dots 27$

- step 4: choose the controller's parameters.

The controller's parameters are chosen as follow:

$$\underline{k} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad Q = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix} \quad P = \begin{pmatrix} 11.5 & 10.5 & 2.5 \\ 10.5 & 23 & 6.5 \\ 2.5 & 6.5 & 3 \end{pmatrix}$$

$$\gamma_f = 1.5 \times (f^U - f_L) = 6 \quad \gamma_g = 1.5 \times (g^U - g_L) = 0.3$$

$$\gamma_\omega = 0.1 \quad \sigma_\omega = 0.3 \quad \omega_0 = 0.01 \quad \omega_{\max} = 0.2 \quad \varepsilon = 0.01$$

- step 5: design the switching mechanism.

In this application, we choose $E_0 = 0.2$ and $\Delta t_0 = 10s$.

The simulation results are shown in Fig 3.6. It can be seen that the controller successfully controls the state x_1 of the transformed Chua's system. After 15s, the tracking error is smaller than 0.01. Fig 3.6c also shows that the state vector \underline{x} stays in the control region $\Omega_{\underline{x}}$ for all time. Fig 3.6d shows that there is no chattering in the control signal. Moreover, it is interesting to note that the magnitude of the control signal of the proposed controller is much smaller than the one in [13]. From Fig 3.6f, we observe that the controller switches from learning mode to operating mode at around 22s. The number of adaptive parameters reduces from 54 (learning mode) to 1 (operating mode). After 22s, there is no significant degenerate in the tracking performance even that there is only 1 adaptive parameter updated online.

3.5. Conclusion

This chapter has presented an indirect AFC scheme, in which an estimator is used to compensate the approximation error. To increase the computational efficiency, a mechanism has also been proposed to automatically switch the controller from learning mode to operating mode to reduce the number of online adaptive parameters. The stability analysis and required conditions of the proposed control scheme has been derived. Application to an inverted pendulum and a Chua's chaotic circuit shows good tracking result in both modes and the number of online adaptive parameters eventually reduces to 1 in operating mode.

Only indirect AFC has been discussed so far. In next chapter, direct AFC will be discussed and solutions to its limitation will be proposed.

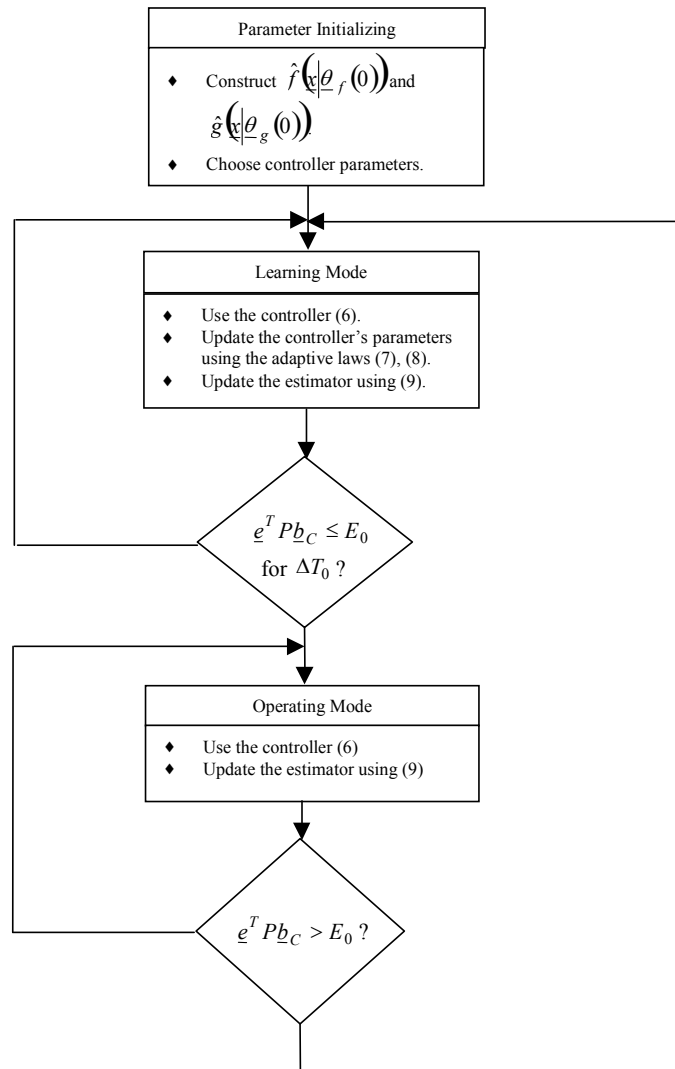


Fig 3.1: The flowchart of the switching mechanism

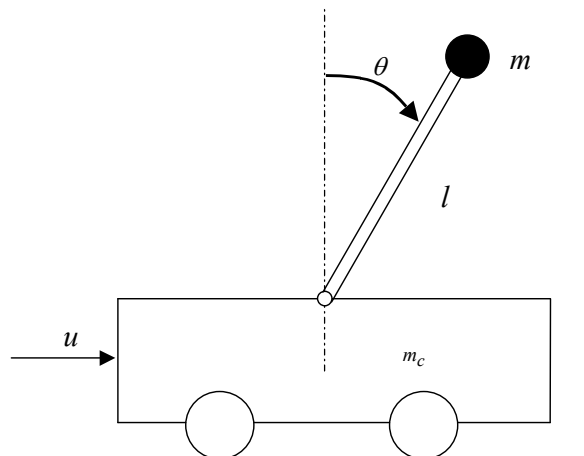
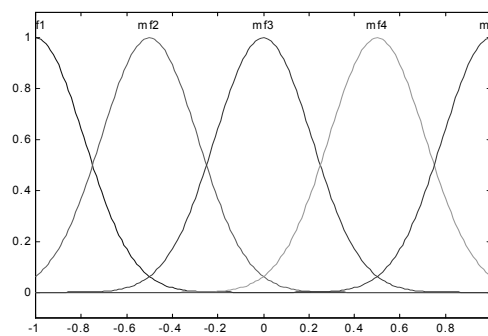
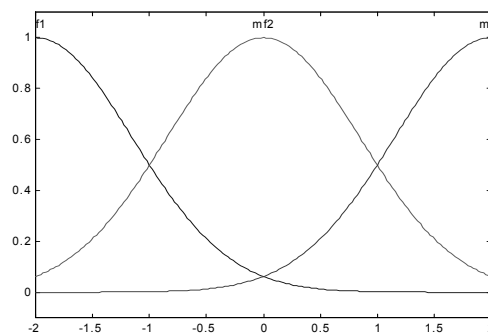
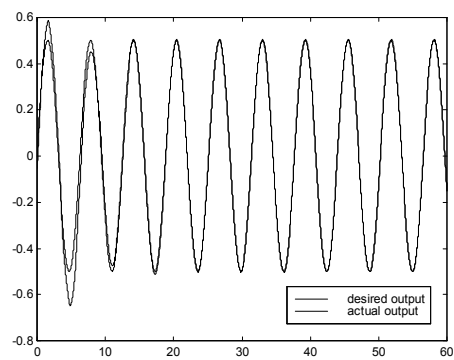
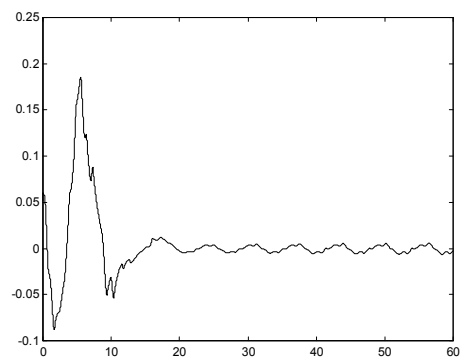
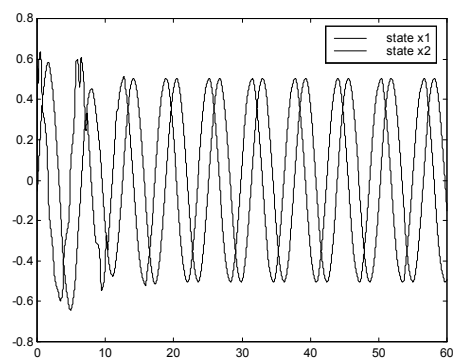
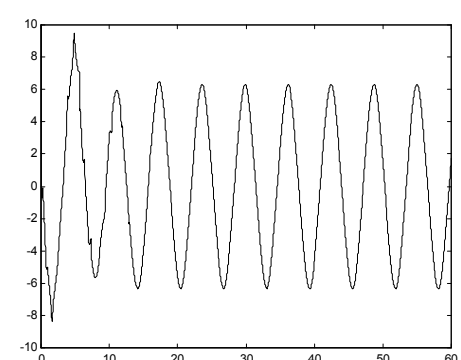
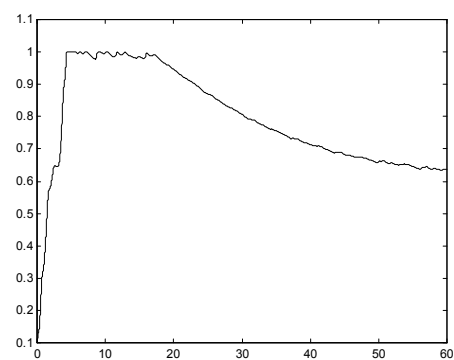
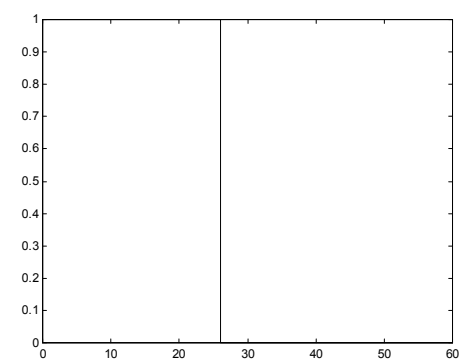


Fig 3.2. The inverted pendulum

Fig 3.3. Membership functions for x_1, x_2 in application 1Fig 3.5. Membership functions for x_1, x_2, x_3 in application 2

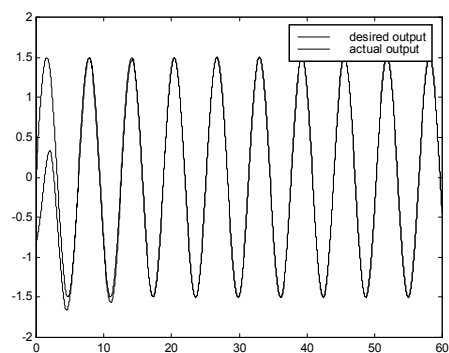


a) Angular position

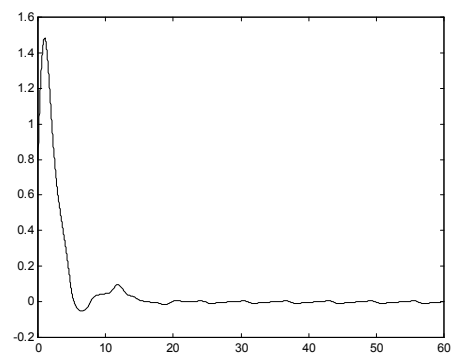
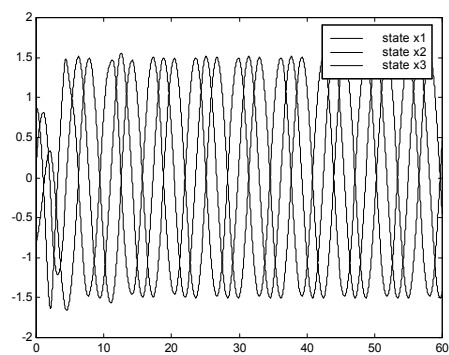
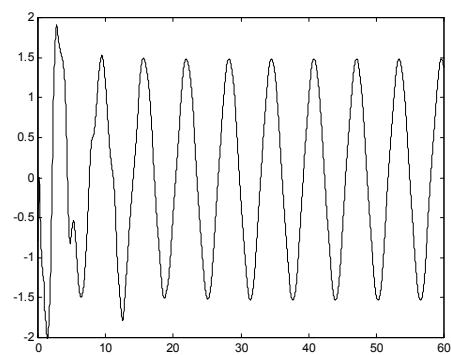
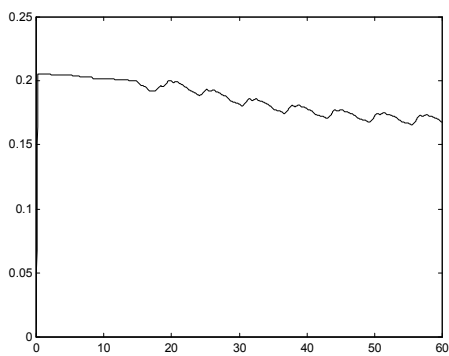
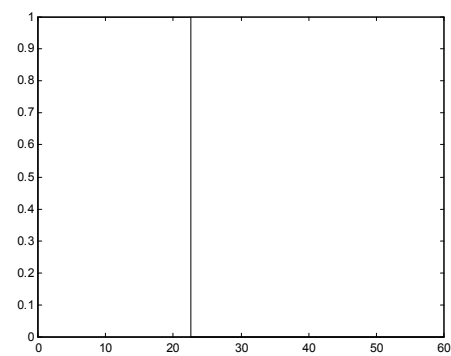
b) Tracking error $y(t) - r(t)$ c) State variable x_1 and x_2 d) Control signal $u(t)$ e) The estimator value $\hat{w}(t)$ 

f) Control mode

Fig 3.4. Simulation results for the inverted pendulum



a) Output

b) Tracking error $y(t) - r(t)$ c) State variable x_1 and x_2 d) Control signal $u(t)$ e) The estimator value $\hat{w}(t)$ 

f) Control mode

Fig 3.6: Simulation results for the Chua's system

4. Chapter 4

DIRECT ADAPTIVE FUZZY CONTROL WITH LESS RESTRICTION ON THE CONTROL GAIN²

4.1. Introduction

In chapters 2 and 3, we have investigated indirect AFC, in which two fuzzy systems are used to model unknown functions $f(\underline{x})$ and $g(\underline{x})$ of the affine nonlinear plant (3.1). Direct AFC, on the other hand, needs only one fuzzy system to approximate the whole ideal control $u^* = \frac{1}{g(\underline{x})}(-f(\underline{x}) + \underline{k}^T \underline{e} + r^{(n)})$. Thus, the main advantage of direct AFC is that its structure is simpler than the one of indirect AFC. However, direct AFC generally requires more restrictions on the control gain. The goal of this chapter is to relax the extra restrictions of direct AFC.

First, section 4.2 gives a survey about the required restrictions and some existing solutions in the literature. Then, a direct AFC scheme with less restriction is proposed in section 4.3 using a simple extension of the universal approximation property. Follow that, application to an inverted pendulum and a magnetic levitation system is given in section 4.4 to demonstrate the proposed control scheme. Finally, some conclusions are summarized in section 4.5

4.2. Literature review

While direct AFC results in a less complicated structure than indirect AFC as it employs only one fuzzy system, the singularity problem in indirect AFC is also completely avoided. However, a literature survey shows that direct AFC schemes usually require more restrictions on the control gain $g(\underline{x})$.

In addition to the controllability condition, some extra restrictions on $g(\underline{x})$ are needed for stability and convergence analysis. [51, 86] require that the control gain $g(\underline{x})$ is known. In [3], $g(\underline{x})$ is assumed to be in the form $g(\underline{x}) = \frac{1}{c} \bar{g}(\underline{x})$ in which

² The content of this chapter has been published in International Journal of Control, Automation, and Systems:

P.A. Phan, and T.J. Gale, "Direct adaptive fuzzy control with less restrictions on the control gain", *International Journal of Control, Automation, and Systems*, Vol 5 No 6 Dec 2007, in press.

$c > 0$ is an unknown scalar constant and $\bar{g}(\underline{x})$ is known. The authors of [1, 5, 12, 31] require that $g(\underline{x})$ is an unknown constant. In [35, 73], the bounds of $g(\underline{x})$ and its first derivative need to be known. In [2], it is assumed that $\frac{\partial g(\underline{x})}{\partial x_n} = 0$, i.e. the control gain does not depend on the state variable x_n .

Recently, some researchers have proposed a number of different approaches to relax the extra constraints on $g(\underline{x})$. Wang CH et al [13, 32] propose a solution, in which the control law does not require extra constraint on $g(\underline{x})$. However, $g(\underline{x})$ still needs to be known to implement the adaptive law. Ge et al [2] propose an approach, in which the extra constraints on $g(\underline{x})$ are relaxed by using a novel integral-type Lyapunov function. The authors later comment that due to the integral operation, this approach is complicated and difficult to use in practice [87]. Leu et al [62] propose a solution in which the nonlinearity of $g(\underline{x})$ is treated as a component of the overall uncertainty and is cancelled using a variable structure control term. Thus, the bound of $g(\underline{x})$ is still needed. Park et al [36] propose an approach in which the implicit function theorem is used to solve the problem. A critical step in their design is to determine a constant c such that $c > \frac{1}{2}g(\underline{x})$, thus knowledge of the upper bound of $g(\underline{x})$ is still necessary.

These constraints present difficulties in practice. For instance, the requirement of constant $g(\underline{x})$ restricts the number of plants that direct AFC can be applied to. The requirement of known $g(\underline{x})$ normally requires tests carried on plants to estimate it. Moreover, it cancels out the main advantage of AFC, that is no mathematical model of plants are required. Even the requirement of known bound of $g(\underline{x})$ is a disadvantage. If a too conservative bound value is chosen, it usually results in undesired control action. Thus, experiments are also needed to determine the bound. These extra experiments add complexity, time and cost to the design of direct AFC.

Why does direct AFC require more restrictions than indirect AFC in the stability analysis? Are those extra restrictions really necessary conditions? Or are they used simply to overcome obstacles in the stability analysis? We identify that the obstacle lies in the statement of the approximation property of fuzzy logic systems. In this chapter, using a simple extension of the universal approximation property, we show

that those extra constraints are actually not needed. Based on this property, the stability analysis of direct AFC can be performed very much like its indirect counterpart.

4.3. Direct adaptive fuzzy control with less restriction

Consider nonlinear system (3.1). **Control objective** is to design an adaptive fuzzy controller such that the closed-loop system must be stable in the sense that all the variables in the closed-loop system must be bounded. And the output $y(t)$ of the system follows a continuous reference signal $r(t) \in C^n$.

Assumption 4.1: $g(\underline{x})$ is continuous and the sign of $g(\underline{x})$ is known for $\underline{x} \in \Omega_{\underline{x}}$, where $\Omega_{\underline{x}}$ is the controllability region.

Since $g(\underline{x}) \neq 0$ (controllable condition of system (4.1)) and $g(\underline{x})$ is continuous for \underline{x} in the controllability region $\Omega_{\underline{x}}$, without loss of generality, it can be assumed that $g(\underline{x}) > 0$ for $\underline{x} \in \Omega_{\underline{x}}$.

Assumption 4.2: Define $\underline{r} = [\dot{r}, \ddot{r}, \ddot{r} \dots, r^{(n-1)}]^T$. We assume that $\|\underline{r}\| \leq r_0$ and $\|r^{(n)}\| \leq r_1$ with known constants $r_0, r_1 > 0$.

The ideal control can be chosen as:

$$u^* = \frac{1}{g(\underline{x})} \left(-f(\underline{x}) + \underline{k}^T \underline{e} + r^{(n)} \right) \quad (4.1)$$

Let $v = \underline{k}^T \underline{e} + r^{(n)}$. (4.1) becomes

$$u^*(\underline{X}) = \frac{1}{g(\underline{x})} (-f(\underline{x}) + v) \quad (4.2)$$

in which $\underline{X} = (\underline{x}^T, v)^T \in \Omega_{\underline{X}}$, $\Omega_{\underline{X}} = \{ \underline{X} | \underline{x} \in \Omega_{\underline{x}}, \|\underline{r}\| \leq r_0, \|r^{(n)}\| \leq r_1 \}$.

To approximate $u^*(\underline{X})$, we employ a fuzzy logic controller in the form (2.7)

$$u = \hat{u}(\underline{X} | \underline{\theta}) = \sum_{j=1}^M \theta_j \zeta_j(\underline{X}) \quad (4.3)$$

in which adaptive parameters are the rule consequents θ_j , $j = 1 \dots M$, and $\underline{\theta} = (\theta_1, \theta_2, \dots, \theta_M)^T$.

Adding and subtracting $g(\underline{x})u^*(\underline{X})$ to (3.1), and after some simple manipulation, we have the error dynamics equation:

$$e^{(n)} = -\underline{k}^T \underline{e} + [g(\underline{x})u^*(\underline{X}) - g(\underline{x})\hat{u}(\underline{X}|\underline{\theta})] \quad (4.4)$$

To continue, we introduce lemma 4.1, which is inspired by the proof of universal approximation property given in [88].

Lemma 4.1. *Given arbitrary $\varepsilon^* > 0$, there exist*

$\underline{\zeta}(\underline{X}) = (\zeta_1(\underline{X}), \zeta_2(\underline{X}), \dots, \zeta_M(\underline{X}))^T$ and an ideal parameter vector

$\underline{\theta}^ = (\theta_1^*, \theta_2^*, \dots, \theta_M^*)^T$ such that*

$$g(\underline{x})u^*(\underline{X}) - g(\underline{x})\hat{u}(\underline{X}|\underline{\theta}) = \sum_{j=1}^M c^j (\theta_j^* - \theta_j) \zeta_j(\underline{X}) + \varepsilon \quad (4.5)$$

where $|\varepsilon| \leq \varepsilon^*$ and c^j are some positive constants.

Proof: is given in appendix 4.A \diamond

Applying lemma 4.1 to (4.4), the error dynamic becomes:

$$e^{(n)} = -\underline{k}^T \underline{e} + \left[\sum_{j=1}^M c^j (\theta_j^* - \theta_j) \zeta_j(\underline{X}) + \varepsilon \right].$$

In the vector form,

$$\dot{\underline{e}} = \Lambda_C \underline{e} + \underline{b}_C \left[\sum_{j=1}^M c^j (\theta_j^* - \theta_j) \zeta_j(\underline{X}) + \varepsilon \right] \quad (4.6)$$

where

$$\Lambda_C = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -k_1 & -k_2 & -k_3 & \dots & -k_n \end{pmatrix}, \quad \underline{b}_C = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}.$$

Since Λ_C is a stable matrix, there exists a unique positive definite symmetric $n \times n$ matrix P which satisfies the Lyapunov equation:

$$\Lambda_C^T P + P \Lambda_C = -Q \quad (4.7)$$

where Q is an arbitrary $n \times n$ positive definite matrix chosen such that $\lambda_{\min}(Q) > 1$.

Assumption 4.3 We can determine the upper and lower bounds of the ideal control signal:

$$u_L \leq u^*(\underline{X}) \leq u_U, \quad \forall \underline{X} \in \Omega_{\underline{X}}.$$

This assumption is not a restriction to the plant. It is a reasonable assumption as, in practice, it is essential to choose an actuator that is capable of performing the

required control action. Later, this assumption will be used to keep the adaptive parameters bounded.

Theorem 4.1 *Given system (3.1) satisfying assumptions 4.1, 4.2, and 4.3, a controller (4.3) with the following adaptive law*

$$\dot{\theta}_j = \begin{cases} \gamma \underline{e}^T P \underline{b}_C \zeta_j(\underline{X}) & \text{if } (u_L < \theta_j < u_U) \\ & \text{or } (\theta_j = u_U \text{ and } \gamma \underline{e}^T P \underline{b}_C \zeta_j(\underline{X}) < 0) \\ & \text{or } (\theta_j = u_L \text{ and } \gamma \underline{e}^T P \underline{b}_C \zeta_j(\underline{X}) > 0) \\ 0 & \text{if } (\theta_j = u_U \text{ and } \gamma \underline{e}^T P \underline{b}_C \zeta_j(\underline{X}) \geq 0) \\ & \text{or } (\theta_j = u_L \text{ and } \gamma \underline{e}^T P \underline{b}_C \zeta_j(\underline{X}) \leq 0) \end{cases} \quad (4.8)$$

where γ is the adaptive gain, will guarantee that:

i. The adaptive parameters are bounded:

$$u_L \leq \theta_j \leq u_U, \quad j = 1 \dots M.$$

ii. The tracking error $\underline{e}(t)$ is bounded by:

$$\|\underline{e}(t)\| \leq \sqrt{\frac{2 \max \left\{ V(0), \frac{1}{2\alpha} \left(\frac{c}{\gamma} + \|P \underline{b}_C\|^2 \|\mathcal{E}^*\|^2 \right) \right\}}{\lambda_{\min}(P)}}, \quad \forall t > 0,$$

in which $\alpha = \frac{(\lambda_{\min}(Q) - 1)}{\lambda_{\max}(P)}$, $V(0)$ is a positive constant dependent on the initial

conditions, and c is a bounded positive constant.

iii. The system is Uniformly Ultimately Bounded (UUB), i.e. $\underline{e}(t)$

converges to compact set $\Omega_{\underline{e}}$ in finite time:

$$\Omega_{\underline{e}} = \left\{ \underline{e}(t) \left| \|\underline{e}(t)\| \leq \sqrt{\frac{\|P \underline{b}_C\|^2 \|\mathcal{E}^*\|^2}{\lambda_{\min}(Q) - 1}} \right. \right\}$$

Proof

i. $u_L \leq \theta_j \leq u_U, \quad j = 1 \dots M.$

From (4.8), it is obvious that $u_L \leq \theta_j \leq u_U, \quad \forall t \geq 0, \quad j = 1 \dots M.$

ii.

Consider the Lyapunov function candidate

$$V = \frac{1}{2} \underline{e}^T P \underline{e} + \frac{1}{2\gamma} \sum_{j=1}^M c^j (\theta_j^* - \theta_j)^2. \quad (4.9)$$

The time derivative of V along the trajectory of (4.6) is:

$$\begin{aligned} \dot{V} &= -\frac{1}{2}\underline{e}^T Q \underline{e} - \frac{1}{\gamma} \sum_{j=1}^M c^j (\theta_j^* - \theta_j) \dot{\theta}_j + \underline{e}^T P \underline{b}_c \left[\sum_{j=1}^M c^j (\theta_j^* - \theta_j) \zeta_j(\underline{X}) + \varepsilon \right] \\ \Leftrightarrow \dot{V} &= -\frac{1}{2}\underline{e}^T Q \underline{e} + \frac{1}{\gamma} \sum_{j=1}^M c^j (\theta_j^* - \theta_j) (\gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) - \dot{\theta}_j) + \underline{e}^T P \underline{b}_c \varepsilon. \end{aligned} \quad (4.10)$$

If we choose the adaptive law (4.8), we have:

- If $(u_L < \theta_j < u_U)$ or $(\theta_j = u_U \text{ and } \gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) < 0)$ or $(\theta_j = u_L \text{ and } \gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) > 0)$:

$$(\theta_j^* - \theta_j) (\gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) - \dot{\theta}_j) = 0.$$

- If $(\theta_j = u_U \text{ and } \gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) \geq 0)$:

$$\dot{\theta}_j = 0. \text{ And as } \theta_j^* \leq u_U = \theta_j, (\theta_j^* - \theta_j) \leq 0.$$

$$\text{Thus, } (\theta_j^* - \theta_j) (\gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) - \dot{\theta}_j) = (\theta_j^* - \theta_j) \gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) \leq 0.$$

- If $(\theta_j = u_L \text{ and } \gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) \leq 0)$: similarly, we have

$$(\theta_j^* - \theta_j) (\gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) - \dot{\theta}_j) = (\theta_j^* - \theta_j) \gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) \leq 0.$$

Therefore, adaptive law (4.8) leads to

$$(\theta_j^* - \theta_j) (\gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) - \dot{\theta}_j) \leq 0 \quad (4.11)$$

Substituting to (4.10) gives:

$$\dot{V} \leq -\frac{1}{2}\underline{e}^T Q \underline{e} + \underline{e}^T P \underline{b}_c \varepsilon. \quad (4.12)$$

Using the fact that

$$-\frac{1}{2}\underline{e}^T Q \underline{e} \leq -\frac{1}{2}\lambda_{\min}(Q)\|\underline{e}\|^2 \text{ where } \lambda_{\min}(Q) \text{ is the minimum eigen value of } Q,$$

$$\text{and } \underline{e}^T P \underline{b}_c \varepsilon \leq \frac{1}{2}\|\underline{e}\|^2 + \frac{1}{2}\|P \underline{b}_c\|^2 \|\varepsilon\|^2 \leq \frac{1}{2}\|\underline{e}\|^2 + \frac{1}{2}\|P \underline{b}_c\|^2 \|\varepsilon^*\|^2,$$

we have

$$\begin{aligned} \dot{V} &\leq -\frac{1}{2}\lambda_{\min}(Q)\|\underline{e}\|^2 + \frac{1}{2}\|\underline{e}\|^2 + \frac{1}{2}\|P \underline{b}_c\|^2 \|\varepsilon^*\|^2 \\ \Leftrightarrow \dot{V} &\leq -\frac{1}{2}(\lambda_{\min}(Q) - 1)\|\underline{e}\|^2 + \frac{1}{2}\|P \underline{b}_c\|^2 \|\varepsilon^*\|^2. \end{aligned} \quad (4.13)$$

Then, the bound of $\underline{e}(t)$ can be derived as follows. As $u_L \leq \theta_j^* \leq u_U$, and $u_L \leq \theta_j \leq u_U$, we have:

$$\frac{1}{2\gamma} \sum_{j=1}^M c^j (\theta_j^* - \theta_j)^2 \leq \frac{1}{2\gamma} \sum_{j=1}^M c^j (u_U - u_L)^2$$

Multiplying by $\frac{(\lambda_{\min}(Q)-1)}{\lambda_{\max}(P)}$ and substituting to (4.13) gives:

$$\begin{aligned} \dot{V} &\leq -\frac{1}{2}(\lambda_{\min}(Q)-1)\|\underline{e}\|^2 - \frac{(\lambda_{\min}(Q)-1)}{\lambda_{\max}(P)} \frac{1}{2\gamma} \sum_{j=1}^M c^j (\theta_j^* - \theta_j)^2 \\ &\quad + \frac{(\lambda_{\min}(Q)-1)}{\lambda_{\max}(P)} \frac{1}{2\gamma} \sum_{j=1}^M c^j (u_U - u_L)^2 + \frac{1}{2} \|P\underline{b}_C\|^2 \|\varepsilon^*\|^2 \\ &\leq -\frac{(\lambda_{\min}(Q)-1)}{\lambda_{\max}(P)} \left(\frac{1}{2} \lambda_{\min}(P) \|\underline{e}\|^2 + \frac{1}{2\gamma} \sum_{j=1}^M c^j (\theta_j^* - \theta_j)^2 \right) \\ &\quad + \frac{(\lambda_{\min}(Q)-1)}{\lambda_{\max}(P)} \frac{1}{2\gamma} \sum_{j=1}^M c^j (u_U - u_L)^2 + \frac{1}{2} \|P\underline{b}_C\|^2 \|\varepsilon^*\|^2 \end{aligned}$$

Let

$$\frac{(\lambda_{\min}(Q)-1)}{\lambda_{\max}(P)} \sum_{j=1}^M c^j (u_U - u_L)^2 = c,$$

$$\text{and } \frac{(\lambda_{\min}(Q)-1)}{\lambda_{\max}(P)} = \alpha,$$

we have:

$$\begin{aligned} \dot{V} &\leq -\alpha V + \frac{1}{2} \left(\frac{c}{\gamma} + \|P\underline{b}_C\|^2 \|\varepsilon^*\|^2 \right) \\ \Leftrightarrow V(t) &\leq e^{-\alpha t} \left[V(0) - \frac{1}{2\alpha} \left(\frac{c}{\gamma} + \|P\underline{b}_C\|^2 \|\varepsilon^*\|^2 \right) \right] + \frac{1}{2\alpha} \left(\frac{c}{\gamma} + \|P\underline{b}_C\|^2 \|\varepsilon^*\|^2 \right). \end{aligned}$$

Thus, $V(t) \leq \max \left\{ V(0), \frac{1}{2\alpha} \left(\frac{c}{\gamma} + \|P\underline{b}_C\|^2 \|\varepsilon^*\|^2 \right) \right\}, \forall t > 0$. From the definition of

V (4.9), the tracking error vector $\underline{e}(t)$, is bounded by:

$$\|\underline{e}(t)\| \leq \sqrt{\frac{2 \max \left\{ V(0), \frac{1}{2\alpha} \left(\frac{c}{\gamma} + \|P\underline{b}_C\|^2 \|\varepsilon^*\|^2 \right) \right\}}{\lambda_{\min}(P)}}, \quad \forall t > 0 \quad (4.14)$$

iii.

Since $\lambda_{\min}(Q) > 1$, equation (4.13) implies that \dot{V} is negative when $\frac{1}{2}(\lambda_{\min}(Q)-1)\|\underline{e}\|^2 \geq \frac{1}{2}\|P\underline{b}_c\|^2\|\varepsilon^*\|^2$. This implies that the system is **UUB**, i.e. $\underline{e}(t)$ converges to compact set Ω_ε in finite time:

$$\Omega_\varepsilon = \left\{ \underline{e}(t) \left| \|\underline{e}(t)\| \leq \sqrt{\frac{\|P\underline{b}_c\|^2\|\varepsilon^*\|^2}{\lambda_{\min}(Q)-1}} \right. \right\}. \quad (4.15)$$

End of proof \diamond

Remark 4.1 To compensate for the approximation error ε^* , some authors have proposed different approaches such as using supervisory control, and error bound estimation, etc. We have proposed use of an approximation error estimator in chapter 3. In this chapter, for clarity, we assume that the approximation error ε is sufficiently small. This assumption becomes more likely with the use of the self-structuring fuzzy system presented in next chapter.

Remark 4.2 It should be noted that, in the literature, there are other modified adaptive laws to guarantee the boundedness of adaptive parameters. One of the most popular approaches is using the σ -modification adaptive law:

$$\dot{\theta}_j = \gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) - \sigma \theta_j. \quad (4.16)$$

However, the design parameter σ does not have a clear physical meaning. It is often chosen as “a small value”, which is ambiguous. The relationship between σ and the bounds of adaptive parameters is not explicit. Even if the adaptive parameters are bounded, it does not guarantee the control signal will stay in the desired range. Here, by utilizing assumption 4.4 and adaptive law (4.8), we guarantee that adaptive parameters are bounded and the control action stays in an explicit range specified by designers.

Remark 4.3 From theorem 4.1.iii, the tracking error can be made arbitrarily small by tuning \underline{k} (to adjust $\|P\underline{b}_c\|$), $\lambda_{\min}(Q)$ and choosing a good approximation structure to keep ε^* small. Larger $\|P\underline{b}_c\|$, $\lambda_{\min}(Q)$ will lead to a smaller tracking error. However, too large $\|P\underline{b}_c\|$, $\lambda_{\min}(Q)$ will result in chattering and high gain control. Therefore, in practical applications, the design parameters should be adjusted carefully for achieving suitable tracking performance and control action.

Remark 4.4 Since the controller is only valid when the state vector \underline{x} is in the desired compact set $\Omega_{\underline{x}}$, it is necessary to keep \underline{x} in $\Omega_{\underline{x}} \forall t \geq 0$. From theorem 4.1.ii, this can be done by choosing sufficiently large γ , small initial condition $V(0)$, and suitable reference signal $r(t)$.

Remark 4.5 Even though the control performance can be tuned intuitively as shown in remarks 4.3 and 4.4, the bounds from theorem 4.ii and 4.iii are very conservative and have no practical use. These bounds depend on $c = \frac{(\lambda_{\min}(Q)-1)}{\lambda_{\max}(P)} \sum_{j=1}^M c^j (u_U - u_L)^2$, which in turn depends on $\sum_{j=1}^M c^j (\theta_j^*)^2 \cdot \sum_{j=1}^M c^j (\theta_j^*)^2$ is unknown and can be arbitrarily large. Therefore, design parameters chosen using the bound of $\sum_{j=1}^M c^j (\theta_j^*)^2$ are very conservative and have no practical use. A survey shows that existing AFC has the same limitation. A future research would be to derive tighter bounds so that design parameters can be selected explicitly to keep system signals in desired compact sets.

4.4. Applications

To demonstrate the theoretical results, we present two applications to an inverted pendulum and a magnet levitation system.

4.4.1. Inverted pendulum

The inverted control problem is given in section 3.4.1. The control objective is to make the angular position $y = x_1$ track the reference signal $r(t) = 0.5 \sin(t)$.

The operating input ranges are chosen as follows:

$$x_1 \in [-1, 1]; \quad x_2 \in [-1, 1]; \quad v \in [-1, 1].$$

The membership functions of each input variable x_1 , x_2 , and v are chosen as shown in figure 3.3. All possible rules are used. Thus, there are $5 \times 5 \times 5 = 125$ rules. All the consequent values are initially chosen as zero.

From remarks 4.3 and 4.4, the design procedure can be:

- choose \underline{k} , and Q .
- estimate P .
- tune γ until satisfied performance is obtained.

In this application, the controller parameters are chosen as follows:

$$\underline{k} = [1 \quad 1]^T; Q = \begin{bmatrix} 20 & 0 \\ 0 & 10 \end{bmatrix}; P = \begin{bmatrix} 25 & 10 \\ 10 & 15 \end{bmatrix}; \gamma = 50;$$

$$u_L = -10; u_U = 10;$$

The results are shown in figures 4.1-4.3. It can be seen that the inverted pendulum is successfully controlled by the direct adaptive fuzzy controller. From an initial tracking error of $-\pi/6$, it converges quickly to the range $[-0.02, 0.02]$. The control signal is always in the range $[u_L, u_U] = [-10, 10]$ as shown in figure 4.3.

The same application is also controlled successfully in [32, 51]. However, Gao [51] requires the determination of $g(\underline{x})^{-1}$. In Wang [32], $g(\underline{x})$ needs to be known to implement the adaptive algorithm (equation 28). Also, the bounds of $f(\underline{x})$ and $g(\underline{x})$ are required.

Here, we have shown that the only requirement on the control gain is its sign. This simplifies the design process and eliminates the time and cost of determining those extra requirements.

4.4.2. Magnetic levitation system

In this application, the control objective is to control the position of a magnet suspended above an electromagnet, where the magnet is constrained so that it can only move in the vertical direction (figure 4.4). The equation of motion of this system is:

$$\ddot{y}(t) = -g + \frac{\alpha}{M} \text{sgn}(i) \frac{i^2(t)}{y(t)} - \frac{\beta}{M} \dot{y}(t)$$

where $y(t)$ is the distance of the magnet above the electromagnet, $i(t)$ is the current flowing in the electromagnet, M is the mass of the magnet, and g is the gravitational constant. The parameter β is a viscous friction coefficient that is determined by the material in which the magnet moves, and α is a field strength constant that is determined by the number of turns of wire on the electromagnet and the strength of the magnet. In this application, we choose $M = 3kg$, $\alpha = 15$, and $\beta = 12$. The desired position $y_d(t)$ is taken randomly in the range $[0.5cm, 4cm]$. The reference trajectory is

generated using a reference model with transfer function $\frac{y_t(s)}{y_d(s)} = \frac{4}{(s+2)(s+2)}$.

Let $x_1 = y(t)$, $x_2 = \dot{y}(t)$, and $u = \text{sgn}(i)i^2(t)$. Thus, the current i can be calculated as $i = \text{sgn}(u)\sqrt{\text{abs}(u)}$. The dynamic equations become

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -g - \frac{\beta}{M}x_2 + \frac{\alpha}{Mx_1}u \\ y &= x_1\end{aligned}$$

which is in the affine form (3.1). Therefore, we can apply our proposed direct AFC to control this system.

The range of the inputs are:

$$x_1 \in [0,5]; x_2 \in [-5,10]; v \in [-10,10].$$

The membership functions of the three input variables are in figure 4.5-4.7. All the consequent values are initially chosen as zero.

Using the same design procedure in application 1, the controller parameters are:

$$\underline{k} = [1 \quad 1]^T; Q = \begin{bmatrix} 20 & 0 \\ 0 & 10 \end{bmatrix}; P = \begin{bmatrix} 25 & 10 \\ 10 & 15 \end{bmatrix}; \gamma = 25;$$

$$u_L = -25; u_U = 25 \text{ (this implies that current } i \text{ is in the range } [-5A, 5A]).$$

The results are shown in figures 4.8-4.10. It can be observed that the actual output tracks closely the reference trajectory. Figure 4.9 shows that the tracking error is maintained in the range $[-0.1cm, 0.1cm]$, and the set-point error converges to a very small neighbourhood of zero. Similar to the first application, the only requirement for the control gain $g(\underline{x})$ is its sign, which is positive in this case. Further knowledge of $g(\underline{x})$ or its bounds are not necessary.

4.5. Conclusion

This chapter has proposed a direct AFC scheme, which requires less restriction. As a result, direct AFC becomes superior compared to indirect AFC as it is simpler in structure, the singularity problem is completely avoided, and no extra restrictions are required. Also, we have proposed a modified adaptive law that not only has more physical meaning than the well-known σ -modification adaptive law, but also guarantees the control action stays in an explicit range.

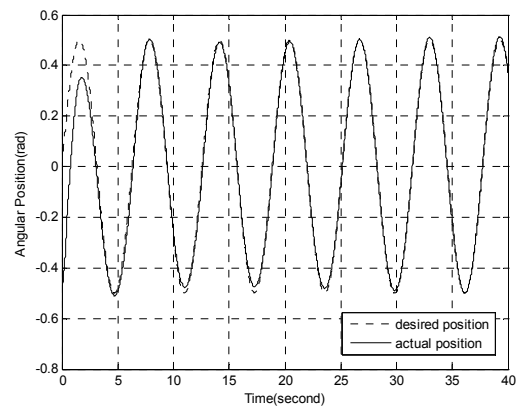


Figure 4.1: angular position

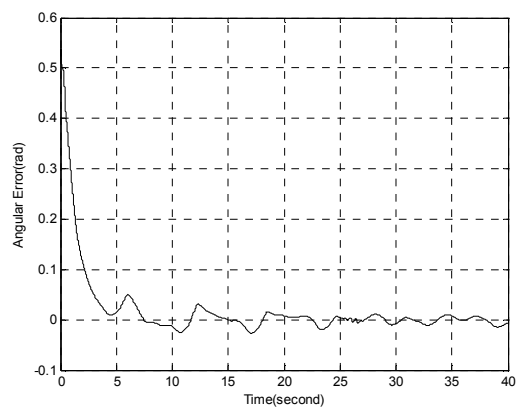


Figure 4.2: tracking error

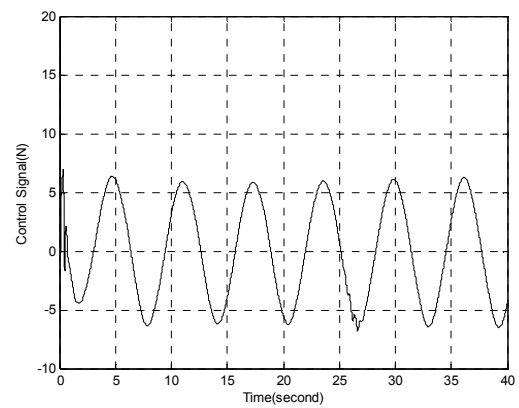


Figure 4.3: control signal

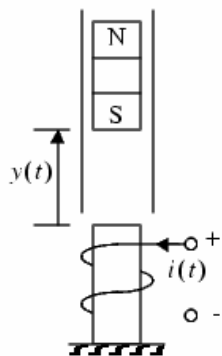
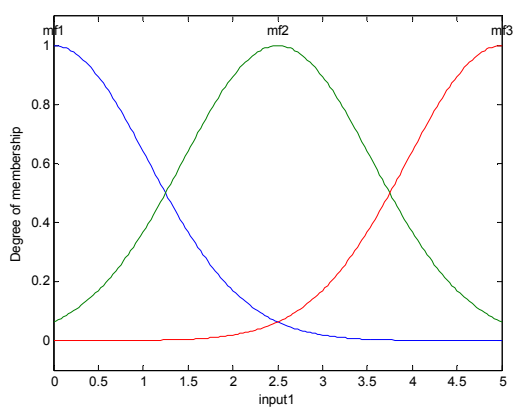
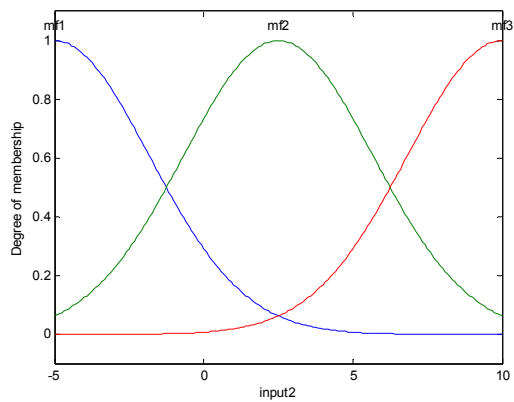


Figure 4.4: a magnet levitation system

Figure 4.5: membership functions for x_1 Figure 4.6: membership functions for x_2

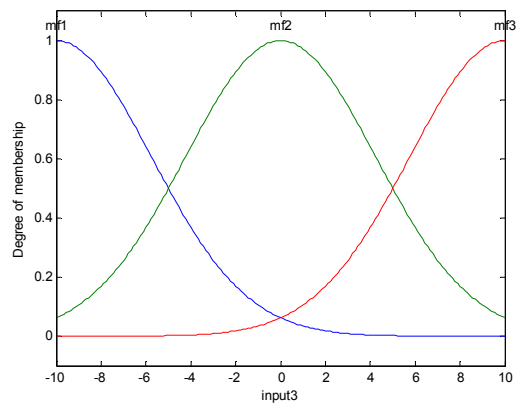
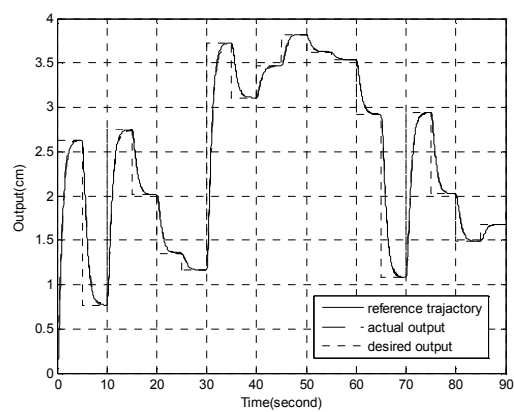
Figure 4.7: membership functions for V 

Figure 4.8: output position (cm)

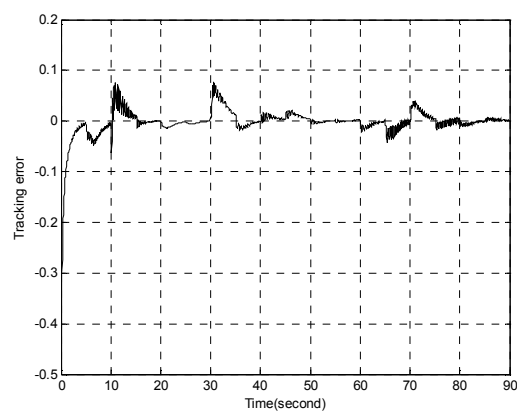


Figure 4.9: tracking error

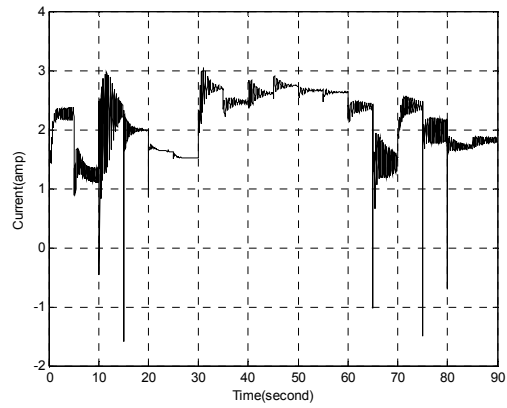


Figure 4.10: control signal

5. Chapter 5

SELF-STRUCTURING DIRECT ADAPTIVE FUZZY CONTROL

5.1. Introduction

In the literature, most AFC schemes employ fuzzy systems with fixed structures. Thus, a designer must specify the number of membership functions and the rule base by trial and error. In many cases, this task is not trivial as exact mathematical models of plants are generally not known. Thus, it is often that the structure used is unnecessarily large or too small to adequately represent the plant. One main objective of the research is to develop an online self-structuring adaptive fuzzy control (SSAFC) scheme.

In this chapter, a SSAFC scheme for affine nonlinear systems is proposed. As a result of the previous chapter, a direct scheme is chosen over an indirect one. First, section 5.2 gives a short survey. Then, section 5.3 presents the direct SSAFC scheme for affine nonlinear systems. This section covers both the description of the self-structuring algorithm and the stability proof. Section 5.4 presents application to an inverted pendulum and a magnetic levitation system. Finally, some conclusions are given in section 5.5.

5.2. Literature review

Self-structuring fuzzy systems require clustering of the input space. Clustering mechanisms include using output error [89, 90], using distance [91, 92], using potential of data points [93], and mountain clustering [94]. There are numerous other algorithms for self-structuring neuro-fuzzy systems, but not all of them are suitable for online control.

Few researchers have proposed self-structuring adaptive fuzzy control [37, 51, 74]. Park et Al [37, 74] propose using a self-structuring fuzzy system, in which rules are added to the rule base as the input space is explored. Triangular membership functions are used. The width of the membership function is pre-defined and unchanged. When one of the input variables moves outside the range of the existing membership functions, a new membership function is created. Then, all the possible rules that are made available by the new membership functions are added to the rule base. This approach eliminates unnecessary rules in regions where the inputs are not

actually explored. However, as the explored input space is evenly divided by the membership functions, there may be redundant rules in regions where the nonlinearity is low and there may be not enough rules in regions where the nonlinearity is high. Another disadvantage is the unrestricted growth of the number of rules.

Gao [51] proposes using a self-organising fuzzy neural system, which is able to add or delete rules from the rule base. The rules are generated based on two criteria, the system error and the ε -completeness of fuzzy rules (ε -completeness of fuzzy rules means that, for any input within the operating range, there exists at least one fuzzy rule such that the firing strength is not less than ε). The rules are pruned based on an error reduction ratio (ERR) concept. Due to the use of the output error for generation of rules, the proposed fuzzy system overcomes the undesirable even distribution of rules in Park et al's approach. However, in our opinion, even if the approach is successful, it is rather complex for online computation as it involves a large matrix calculation in every step and requires memory of all the past input-output data pairs. Also an explicit relationship between error reduction ratio and the number of rules cannot be obtained. Thus, in practice there is no guarantee that the size of the fuzzy system will not exceed the hardware capability. Moreover, use of Gaussian membership functions further adds computational complexity to the system as the number of activated rules at a particular moment of time can not be limited (when triangular membership functions are used, the number of activated rules is smaller or equal to 2^n , in which n is the number of input variables).

Stability is an important aspect in control. However, in [37, 51, 74], only stability when the structure is fixed is proved. The stability when the structure is changed has not been shown.

In this chapter, we propose a novel self-structuring direct adaptive fuzzy control (SSDAFC) for affine nonlinear systems, which has the following features:

- Rules are added based on the system error and the ε -completeness of fuzzy rules: thus, our approach overcomes the undesirable even distribution of rules in Park et al's approach.
- To limit the number of rules from growing indefinitely, we propose a simple algorithm to replace membership functions (instead of adding more membership functions) so that the number of rules never exceeds a

predefined upper bound. Our approach avoids using the ERR concept, thus, avoiding large matrix computation and storage of past data.

- To further reduce the computational complexity and increase the interpretability of fuzzy systems, we employ triangular membership functions and allow at most 2 membership functions activated in each input dimension.
- The stability is proved both when the structure is fixed and when the structure is changed

5.3. Self-structuring direct adaptive fuzzy control for affine nonlinear systems

Beside assumptions 4.1, 4.2, and 4.3, to propose the SSDAFC for affine nonlinear systems (4.1), an additional assumption is needed.

Assumption 5.1 We can determine the upper bound B_{rule} of the required number of rules that achieves the desired approximation accuracy.

This assumption is reasonable, as in practice it is important to select computational hardware that is capable of implementing the controller. This assumption is used to ensure that the controller does not exceed the hardware capacity. Also, this assumption is less restrictive than the assumption required in fixed-structured AFC that “Designers are able to construct a fuzzy rule base that achieves the desired approximation accuracy”. Knowing the upper bound B_{rule} of the required number of rules, the self-structuring algorithm will automatically construct a satisfactory rule base.

Let $\underline{\theta} = (\theta_1, \theta_2, \dots, \theta_N)^T$ be the adaptive parameter vector of the final fuzzy controller. From assumption 5.1, we have $N \leq B_{rule}$. Thus, $\underline{\theta} = \begin{pmatrix} \underline{\theta}_{ac}^T & \underline{\theta}_{in}^T \end{pmatrix}^T$ in which $\underline{\theta}_{ac} = (\theta_1, \theta_2, \dots, \theta_M)^T$, ($M \leq N$) is the vector of adaptive parameters already activated, and $\underline{\theta}_{in} = (\theta_{M+1}, \theta_{M+2}, \dots, \theta_N)^T$ is the vector of adaptive parameters not yet generated (inactivated). It should be noted that $\underline{\theta}_{in}$ is unknown and only required for analytical purposes. The control signal is chosen as:

$$u = \hat{u}(\underline{X}|\underline{\theta}_{ac}) = \sum_{j=1}^M \theta_j \zeta_j(\underline{X}). \quad (5.1)$$

5.3.1. Description of the self-structuring algorithm

The key roles of the online self-structuring algorithm include:

- Decide when the structure needs to change.
- Decide whether a new membership function should be added or an old membership function should be replaced.
- Determine the values of membership function parameters and initial values of the rule consequents.

The flowchart of the algorithm is given in Fig 5.1.

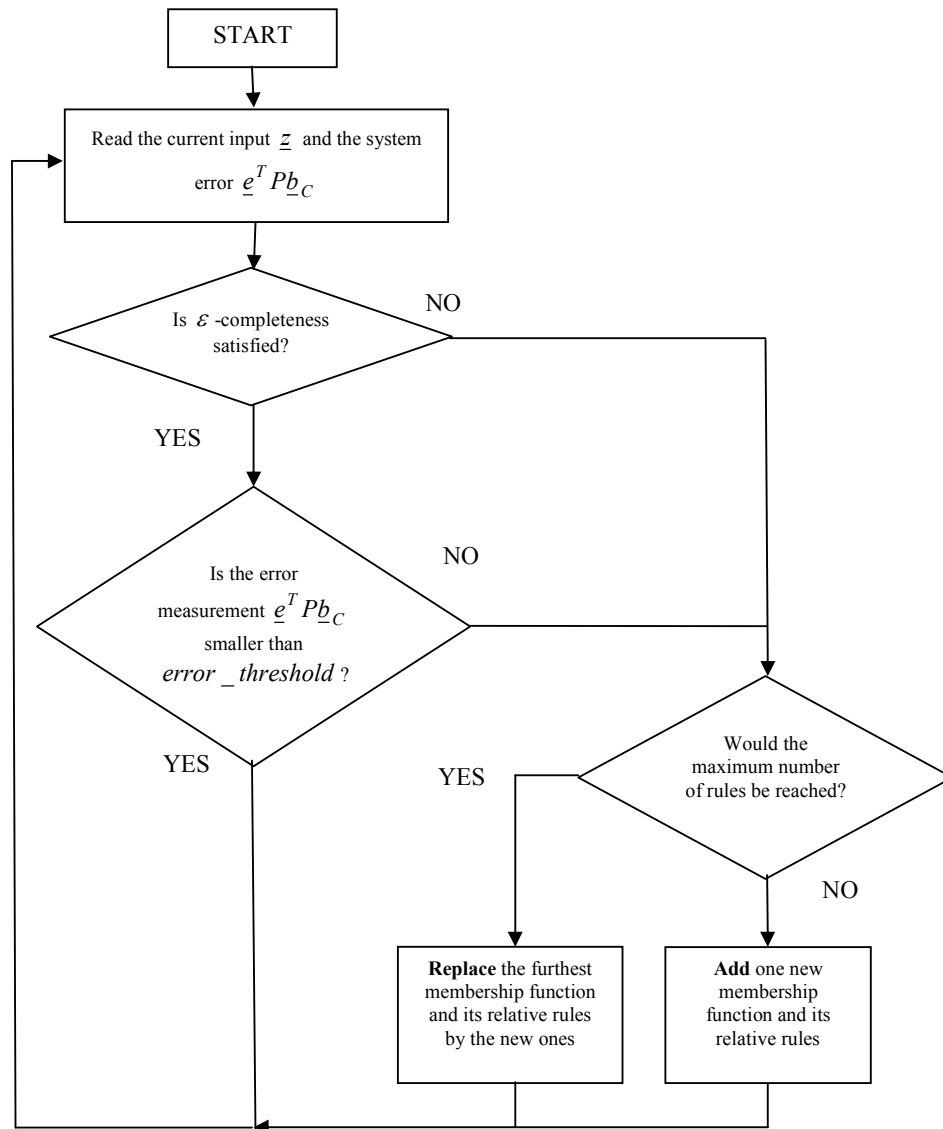


Fig 5.1: self-structuring algorithm flowchart

5.3.1.1. Criteria for rule generation

Two criteria for rule generation are system error and ε -completeness:

- System error:

$\underline{e}^T P \underline{b}_C$ represents the system error. In the adaptive law, the rule consequents are adjusted to reduce $\underline{e}^T P \underline{b}_C$. When $\underline{e}^T P \underline{b}_C = 0$, the output error is zero, and the rule consequents do not need to change. Therefore, when $\underline{e}^T P \underline{b}_C$ is equal to or larger than a predefined value *error_threshold*, a new membership function is considered.

- The ε -completeness:

In Gao [9], ε -completeness of fuzzy rules is defined as “for any input within the operating range, there exists at least one fuzzy rule such that the match degree (or firing strength) is not less than ε ”. To guarantee the ε -completeness, we make sure that: for any input within the operating range, in every input dimension, there exists at least one membership function such that the membership degree is not less than ε_0 . The relationship between ε and ε_0 is $\varepsilon = \varepsilon_0^n$, where n is the number of inputs. The value of ε_0 is usually selected as $\varepsilon_0 = 0.5$.

If one of the two criteria for rule generation is not satisfied, a new membership function is considered. The algorithm then checks if B_{rule} would be exceeded if the new membership function is added. If the answer is “no”, a new membership function will be added. If the answer is “yes”, an old membership function will be replaced.

5.3.1.2. Adding a membership function and its related rules when the ε -completeness is not satisfied

When the ε -completeness is not satisfied, and B_{rule} will not be reached, a new membership function will be added.

Identify the input dimension to which the new membership function is added. Since the ε -completeness is not satisfied, there is an input dimension that there is no membership function with membership degree greater or equal to ε_0 . The new membership function is added to this input dimension.

Determine the parameters of the new membership function. Parameters of a triangular membership function include its center, left point, and right point. When a new membership function is added, its center is chosen as the current value of the input variable. The left and right points are chosen as the centers of the left and right

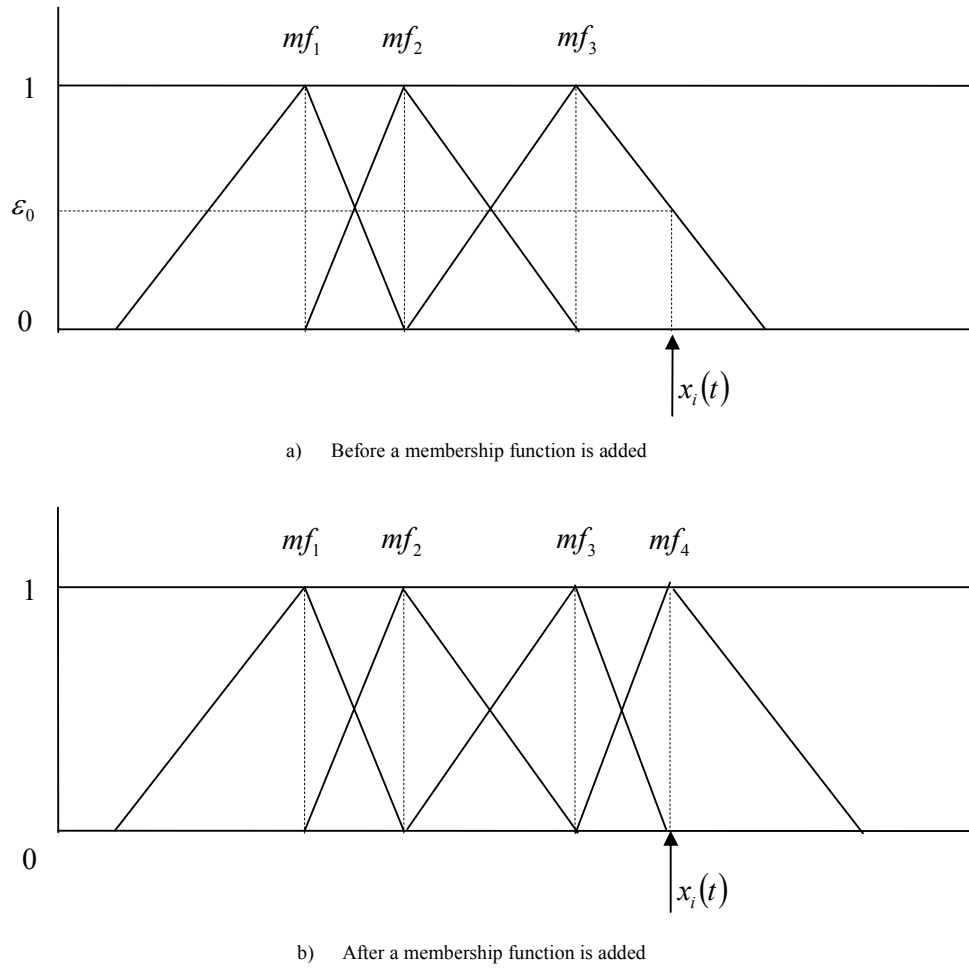


Figure 5.2: if \mathcal{E} -completeness is not satisfied and B_{rule} will not be reached

neighbouring membership functions respectively. In cases when there is no left (or right) neighbouring membership function, the left (or right) point is chosen such as the distance to the center is equal to a predefined value (**max_mf_distance**). Thus, **max_mf_distance** defines the maximum allowed distance between two neighbouring membership functions.

To avoid membership functions being too close, a membership function is only added when the distances between its center and the centers of the neighbouring membership functions are greater than or equal to a predefined value (**min_mf_distance**). Thus, **min_mf_distance** defines the minimum allowed distance between two neighbouring membership functions.

To ensure that there are at most 2 membership functions activated at any time, the neighbouring membership functions are also modified accordingly. The right point of the left neighbouring membership function is modified to the center of new

membership function. The left point of the right neighbouring membership function is also modified to the center of the new membership function.

The new rules and the consequent values are determined as follows. All possible rules made by the new membership function are generated. Since our proposed fuzzy system is an unevenly-distributed grid-type, when a membership function is added, 2^{n-1} new rules are made possible where n is the number of inputs. All the new rules' consequents are, then initialized to the current output of the fuzzy system.

Fig 5.2 illustrates how a membership function is added in this case. Fig 5.2a shows that the membership degree is less than ε_0 . Thus, a membership function mf_4 is added at $x_i(t)$ as shown in fig 5.2b. It can be seen that the left point of mf_4 is chosen as the center of mf_3 . Since there is no membership function on the right of mf_4 , the right point of mf_4 is chosen as $x_i(t) + \text{max_mf_distance}$. The right point of mf_3 is modified to the center of new membership function mf_4 .

5.3.1.3. Replacing a membership function and its related rules when the ε -completeness is not satisfied

When the ε -completeness is not satisfied, and B_{rule} will be reached, a membership function will be replaced.

The new membership function and its related rules are determined the same way in section 5.3.1.2.

The membership function to be removed is determined as follows. In the input dimension to which the new membership function is added, the algorithm searches for the furthest membership function from the current point. That furthest membership function is the membership function to be removed. All rules related to the removed membership function are also deleted from the rule base.

Fig 5.3 demonstrates how an old membership function is replaced in this case. Fig 5.3a shows the membership functions before a membership function is replaced. It can be seen that membership function mf_1 is the furthest membership function from $x_i(t)$. Thus, it will be replaced by a new one. Fig 5.3b shows the old membership function mf_1 (in fig 5.3a) is replaced by the new membership function mf_1 . The center of the new mf_1 is chosen as $x_i(t)$. The left point of mf_1 is chosen as x_3 . Since there is no membership function on the right of new mf_1 , the right point of mf_1 is

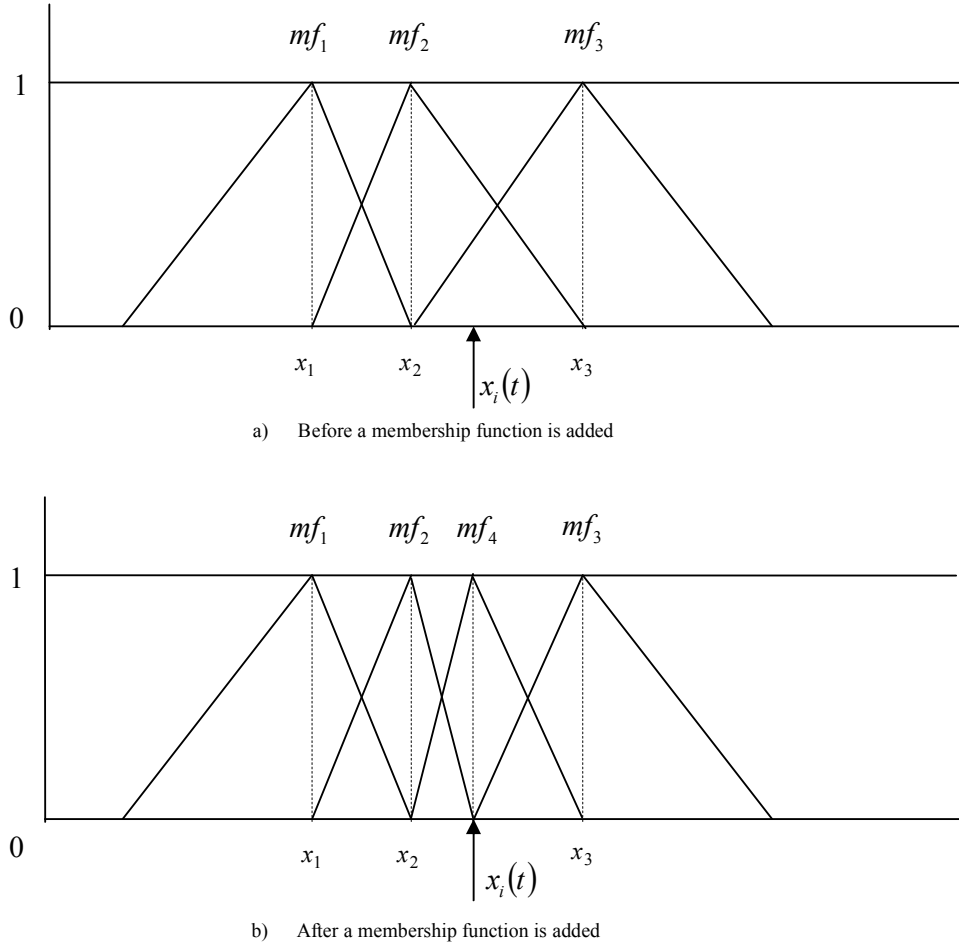


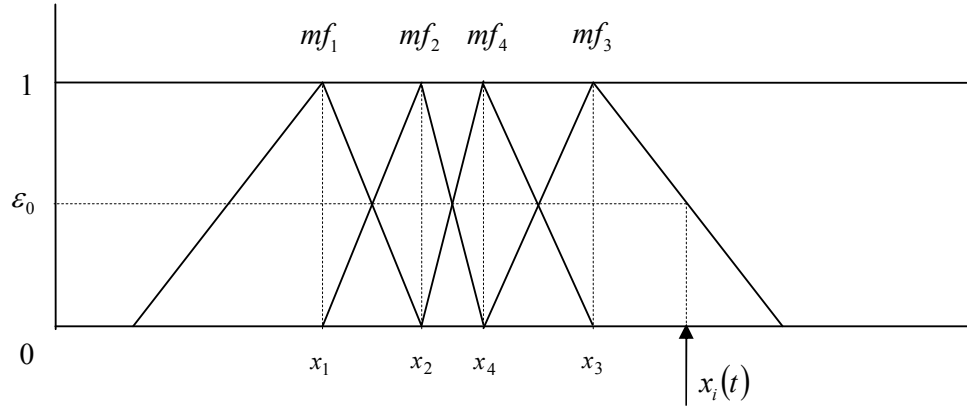
Figure 5.3: if $\underline{e}^T P \underline{b}_C \geq \text{error_threshold}$, distance between $x_i(t)$ and the closest membership function center $\geq \text{distance_threshold}$, and B_{rule} will not be reached

chosen as $x_i(t) + \text{max_mf_distance}$. Since there is no membership function on the left of mf_2 now, its left point is modified to $x_i(t) - \text{max_mf_distance}$.

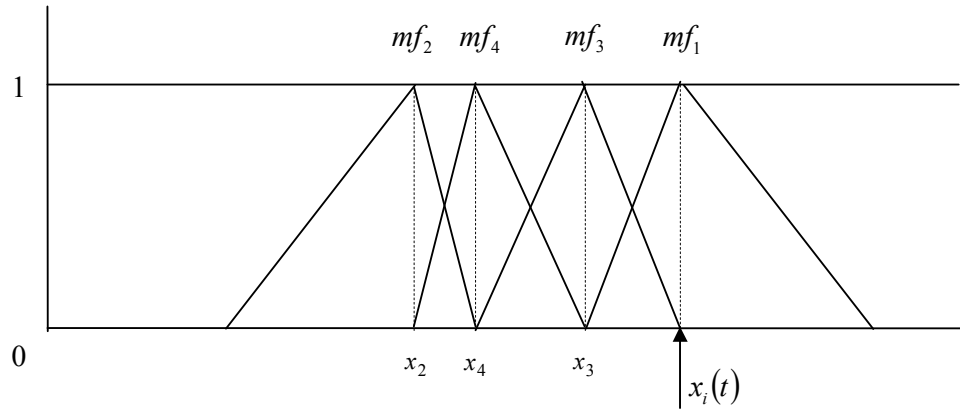
5.3.1.4. Adding a membership function and its related rules when $\underline{e}^T P \underline{b}_C$ is equal to or larger than error_threshold

When $\underline{e}^T P \underline{b}_C \geq \text{error_threshold}$, and B_{rule} will not be reached, an old membership function will be replaced.

Identify the input dimension to which the new membership function is added. The following procedure is used. The rule with maximum firing strength at that moment is selected. Then, the new membership function is added to the input with the maximum membership function degree. The reason is that the large system error



a) Before a membership function is added



b) After a membership function is added

Figure 5.4: if \mathcal{E} -completeness is not satisfied and B_{rule} will be reached

indicates that membership functions in that input are not sufficient to represent the nonlinearity in the region.

The rest of the procedure is the same as the one described in section 5.3.1.2.

Fig 5.4 demonstrates how a new membership function is added in this case. Fig 5.4a shows the membership functions before a new membership function is added. Distance between $x_i(t)$ and the closest membership function center in this case is $|x_i(t) - x_2|$ ($\geq \text{min_mf_distance}$). As shown in fig 5.4b, membership function mf_4 is added at $x_i(t)$. The left point of mf_4 is chosen as x_2 . The right point of mf_4 is chosen as x_3 . The neighbouring membership functions (mf_2 and mf_3) are also modified. The right point of mf_2 and the left point of mf_3 are modified to the center of mf_4 .

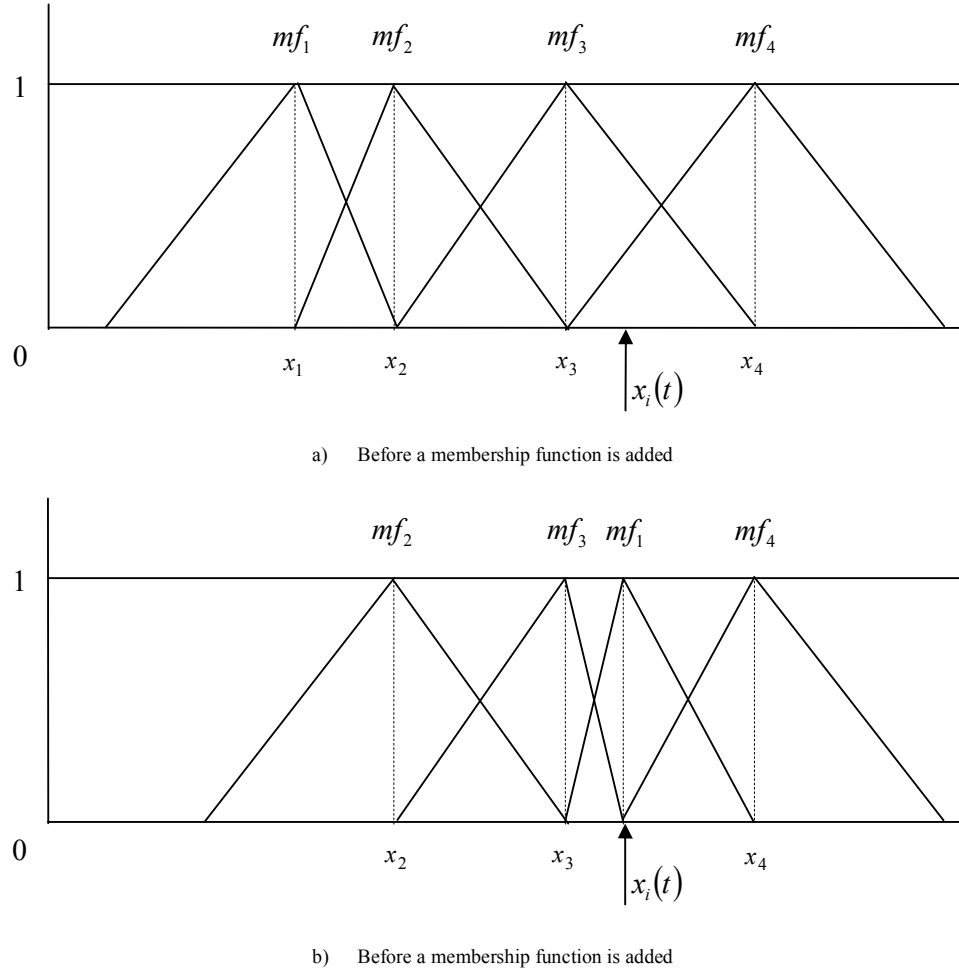


Figure 5.5: if $\underline{e}^P P \underline{b}_C \geq error_threshold$, distance between $x_i(t)$ and the closest membership function center $\geq distance_threshold$, and B_{rule} will be reached

5.3.1.5. Replacing a membership function and its related rules when the error measurement $\underline{e}^T P \underline{b}_C$ is equal to or larger than $error_threshold$

When $\underline{e}^T P \underline{b}_C \geq error_threshold$, and B_{rule} will be reached, a new membership function will be added.

The new membership function and its related rules are determined the same way in the section 5.3.1.4.

The old membership function to be replaced is determined the same way as in section 5.3.1.3.

Fig 5.5 demonstrates how an old membership function is replaced in this case. Fig 5.5a shows the functions before a membership function is replaced. Distance between $x_i(t)$ and the closest membership function center is $|x_i(t) - x_3|$

($\geq \mathbf{min_mf_distance}$). It can be seen that mf_1 is the furthest membership function from $x_i(t)$. Thus, it will be replaced. Fig 5.5b shows the new membership function mf_1 and the modified membership functions mf_3 and mf_2 . The center of mf_1 is chosen as $x_i(t)$. The left point of mf_1 is chosen as x_3 . The right point of mf_1 is chosen as $x_i(t) + \mathbf{max_mf_distance}$. The right point of mf_3 is modified to the center of mf_1 . The left point of mf_2 is modified to $x_i(t) - \mathbf{max_mf_distance}$.

5.3.1.6. Parameters

The self-structuring algorithm has four design parameters. ε_0 defines the completeness of fuzzy rules, *error_threshold* defines the minimum level of error to trigger structure change, **min_mf_distance** defines the minimum allowed distance between two neighbouring membership functions, and **max_mf_distance** defines the maximum allowed distance between two neighbouring membership functions.

Therefore, using larger values of ε_0 or smaller values of *error_threshold*, **min_mf_distance**, or **max_mf_distance** will result in more rules being generated. However, the number of rules is always bounded by B_{rule} .

5.3.2. SSDAFC

The stability of the SSDAFC is given in the following theorem.

Theorem 5.1 *Given system (3.1) satisfying assumptions 4.1, 4.2, 4.3, and 5.1, a controller (5.1) with the self-structuring algorithm described in section 5.3.1 and the adaptive law*

$$\dot{\theta}_j = \begin{cases} \gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) & \text{if } (u_L < \theta_j < u_U) \\ & \text{or } (\theta_j = u_U \text{ and } \gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) < 0) \\ & \text{or } (\theta_j = u_L \text{ and } \gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) > 0) \\ 0 & \text{if } (\theta_j = u_U \text{ and } \gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) \geq 0) \\ & \text{or } (\theta_j = u_L \text{ and } \gamma \underline{e}^T P \underline{b}_c \zeta_j(\underline{X}) \leq 0) \end{cases}$$

will guarantee that:

iv. The adaptive parameters are bounded:

$$u_L \leq \theta_j \leq u_U, \quad j = 1 \dots M.$$

v. The tracking error $\underline{e}(t)$ is bounded by:

$$\|\underline{e}(t)\| \leq \sqrt{\frac{2 \max \left\{ V(0), \frac{1}{2\alpha} \left(\frac{c}{\gamma} + \|P\underline{b}_c\|^2 \|\mathcal{E}^*\|^2 \right) \right\}}{\lambda_{\min}(P)}}, \forall t > 0, \quad (5.2)$$

in which $\alpha = \frac{(\lambda_{\min}(Q)-1)}{\lambda_{\max}(P)}$, $V(0)$ and c are bounded positive constants.

- vi. The system is Uniformly Ultimately Bounded (UUB), i.e. $\underline{e}(t)$ converges to compact set $\Omega_{\underline{e}}$ in finite time:

$$\Omega_{\underline{e}} = \left\{ \underline{e}(t) \left| \|\underline{e}(t)\| \leq \sqrt{\frac{\|P\underline{b}_c\|^2 \|\mathcal{E}^*\|^2}{\lambda_{\min}(Q)-1}} \right. \right\}. \quad (5.3)$$

Proof

In theorem 4.1, we have proved the stability of fixed-structured systems. Here, we also need to show the stability when the structure changes. If the Lyapunov function is chosen as in (4.9), it changes when the structure changes. Thus, it is rather difficult to show the stability. To overcome this problem, we choose a new Lyapunov function that also includes the not-yet-generated adaptive parameters:

$$V = \frac{1}{2} \underline{e}^T P \underline{e} + \frac{1}{2\gamma} \sum_{j=1}^M c^j (\theta_j^* - \theta_j)^2 + \frac{1}{2\gamma} \sum_{k=M+1}^N c^k (\theta_k^* - \theta_k)^2, \quad (5.4)$$

in which the values of not yet generated parameters $\theta_k, k = M+1, \dots, N$, are chosen as their initialized values (these values are unknown and only required for analytical purpose). Since P is positive definite and $c^j > 0, j = 1 \dots M, c_k > 0, k = M+1, \dots, N$, it is obvious that $V \geq 0$.

The stability analysis has two steps. First, we show the stability when the structure is fixed. Then, we show that the system is stable at the time the structure changes.

5.3.2.1. When the structure is fixed

From the adaptive law (4.8), it is obvious that theorem 5.1.i holds.

When the structure is fixed, M is unchanged. Using the fact that $\dot{\theta}_j^* = 0, j = 1 \dots M$, and $\dot{\theta}_k^* = \dot{\theta}_k = 0, k = M+1, \dots, N$, the time derivative of V along the trajectory of (4.6) is:

$$\dot{V} = -\frac{1}{2}\underline{e}^T Q \underline{e} - \frac{1}{\gamma} \sum_{j=1}^M c^j (\theta_j^* - \theta_j) \dot{\theta}_j + \underline{e}^T P \underline{b}_c \left[\sum_{j=1}^M c^j (\theta_j^* - \theta_j) \zeta_j(X) + \varepsilon \right]$$

$$\Leftrightarrow \dot{V} = -\frac{1}{2}\underline{e}^T Q \underline{e} + \frac{1}{\gamma} \sum_{j=1}^M c^j (\theta_j^* - \theta_j) \left(\gamma \underline{e}^T P \underline{b}_c \zeta_j(X) - \dot{\theta}_j \right) + \underline{e}^T P \underline{b}_c \varepsilon.$$

This equation is exactly the same as (4.10). Following the same procedure as in theorem 4.1 (equations (4.10) to (4.15)), we have theorem 5.1.ii and 5.1.iii hold.

5.3.2.2. When the structure changes

Now, to guarantee the stability of the system at all time, we need to show that the system is stable when the structure changes.

This can be proved by showing that $V(t)$ defined in (5.4) does not change when the structure changes. Let t_c be the time that the structure changes and M_1 , M_2 be the old and new numbers of rules respectively ($M_1 < M_2$). We will show that $V(t_c^-) = V(t_c^+)$.

With the proposed self-structuring algorithm, the control signal is continuous at t_c :

$$u(t_c^-) = u(t_c^+). \quad (5.5)$$

Given system (3.1), (5.5) leads to $\underline{x}(t_c^-) = \underline{x}(t_c^+)$ and

$$\underline{e}(t_c^-) = \underline{e}(t_c^+). \quad (5.6)$$

From the adaptive law (4.8), (5.6) leads to

$$\theta_j(t_c^-) = \theta_j(t_c^+), \quad j = 1 \dots M_1. \quad (5.7)$$

Moreover, as we chose the values of inactivated adaptive parameters as the values when they are activated, their values do not change at t_c . Thus, we have:

$$\theta_j(t_c^-) = \theta_j(t_c^+), \quad j = (M_1 + 1), \dots, M_2 \quad (5.8)$$

From (5.6), (5.7) and (5.8), we have:

$$\begin{aligned} V(t_c^-) &= \frac{1}{2} \underline{e}(t_c^-)^T P \underline{e}(t_c^-) + \frac{1}{2\gamma} \sum_{j=1}^{M_1} c^j (\theta_j^* - \theta_j(t_c^-))^2 + \frac{1}{2\gamma} \sum_{k=M_1+1}^N c^k (\theta_k^* - \theta_k(t_c^-))^2 \\ &= \frac{1}{2} \underline{e}(t_c^+)^T P \underline{e}(t_c^+) + \frac{1}{2\gamma} \sum_{j=1}^{M_2} c^j (\theta_j^* - \theta_j(t_c^+))^2 + \frac{1}{2\gamma} \sum_{k=M_2+1}^N c^k (\theta_k^* - \theta_k(t_c^+))^2 \\ &= V(t_c^+) \end{aligned}$$

End of proof \diamond

Remark 5.1 All remarks in chapter 4 are also valid for this SSDAFC scheme.

Remark 5.2 It should be noted that the structures generated by the self-structuring algorithm are not the optimal ones. Our goal is not to find the optimal solution, but to find a structure such that all variables are bounded (including the size of the fuzzy controller) and the output follows the reference signal. The proposed self-structuring algorithm satisfies this goal.

Remark 5.3 The main limitation of our approach is that it suffers from “the curse of dimensionality”, the complexity increases exponentially with the number of inputs. This is the trade-off for interpretability. Future research would be to develop a SSDAFC scheme for high-order systems, in which simplicity is critical and interpretability is less important.

5.4. Examples

5.4.1. Inverted pendulum

To demonstrate the proposed controller, its application to the inverted pendulum given in section 3.4.1 is presented. The control objective is to make the angular position $y = x_1$ track the reference signal $r(t) = 0.5 \sin(t)$.

The operating variable ranges are chosen as follows:

$$x_1 \in [-1, 1]; \quad x_2 \in [-1, 1]; \quad v \in [-1, 1].$$

The controller parameters are chosen as follows:

$$\underline{k} = [1 \quad 1]^T; \quad Q = \begin{bmatrix} 20 & 0 \\ 0 & 10 \end{bmatrix}; \quad P = \begin{bmatrix} 25 & 10 \\ 10 & 15 \end{bmatrix}; \quad \gamma = 50$$

$$u_L = -10; \quad u_U = 10.$$

To test the algorithm with different parameters, we perform simulations with 3 different setups as follows. As the fuzzy system has 3 premise variables, *distance_threshold* and **max_mf_distance** are vectors with 3 elements. The maximum allowed distance between 2 membership functions is chosen as half of the input range, i.e **max_mf_distance** = [1 1 1].

	ε_0	<i>error_threshold</i>	min_mf_distance	max_mf_distance	B_{rule}
Setup1	0.5	0.5	[0.2 0.2 0.2]	[1 1 1]	100
Setup2	0.5	0.5	[0.4 0.4 0.4]	[1 1 1]	100

Setup3	0.5	0.5	[0.4 0.4 0.4]	[1 1 1]	20
--------	-----	-----	---------------	---------	----

The initial fuzzy system has only 1 rule (initialized to 0) with 1 membership function in each input dimension as shown in fig 5.6.

The simulation results of setup 1 are shown in figures 5.7. A variable called *self-structuring flag* is used to indicate when the self-structuring performs. When the self-structuring flag switches from 1 to -1 or -1 to 1, it indicates a change of the fuzzy system structure has occurred. It can be observed that the controller successfully controls the inverted pendulum to track the sinusoidal signal $r(t) = 0.5\sin(t)$ (Fig 5.7a). After 30s, the tracking error is as small as in the range $[-0.01, +0.01]$ (Fig 5.7b). The control signal is always in the desired range $[-10, +10]$ (Fig 5.7c). Self-structuring happens in the first 10s (Fig 5.7d). The final fuzzy system has 96 rules, and the bound $B_{rule} = 100$ is never exceeded. The final membership functions in each input dimension are given in Figures 5.8.

The minimum allowed distance between two neighbouring membership functions is defined by **min_mf_distance**. Thus, increasing **min_mf_distance** will result in fewer rules. This is confirmed by simulation results of setup 2. After 30s, the tracking error has reduced to within the range $[-0.02, +0.02]$. The control signal is always in the desired range $[-10, +10]$. The final fuzzy system has 36 rules.

To test how the algorithm replaces membership functions and rules, we change B_{rule} to 20 in setup 3. The results are shown in figures 5.9. It can be seen that the tracking performance is very good. The tracking error is in the range $[-0.02, +0.02]$ after 20s. The number of rules increases quickly in the first 5s to 18 rules. After that, the number of rules never exceeds B_{rule} , i.e. 20 rules. It can be observed that the algorithm replaces membership functions roughly at approximately 10, 14, 17, 32.5, 36, 45 and 54s. At those moments, there is no degradation in tracking performance. This confirms that replacing membership functions does not affect the performance of the control system. The membership functions of the fuzzy system at $t = 60s$ are given in Figures 5.10.

The transient error shown in Fig 5.7c is better than the transient error shown in Fig 5.9c. The reason is explained as follows. When a membership function (and its corresponding rules) is added, the adaptive algorithm needs to make a large

adjustment due the initial error of the newly added adaptive parameters. As a result, the control action changes relatively quick. Thus, the control performance temporarily deteriorates. In Fig 5.9c, a higher number of rules is allowed. Thus, at the start of the simulation, more membership functions are added due to large initial error. As a consequence, performance in this case deteriorates more. When the structures of the fuzzy systems are more stable (after 50s), the errors in both cases are similar.

5.4.2. Magnetic levitation

The magnetic levitation system is given in section 4.4.2. Now, we apply the proposed DSAFC to control this system.

The variable ranges are:

$$x_1 \in [0,5]; x_2 \in [-5,10]; v \in [-10,10].$$

The controller parameters are:

$$\underline{k} = [1 \ 1]^T; Q = \begin{bmatrix} 20 & 0 \\ 0 & 10 \end{bmatrix}; P = \begin{bmatrix} 25 & 10 \\ 10 & 15 \end{bmatrix}; \gamma = 50.$$

$$u_L = -25; u_U = 25 \text{ (this implies that current } i \text{ is in the range } [-5A, 5A]).$$

To test the algorithm with different parameters, we perform simulations with 3 different setups as follows:

	ε_0	<i>error_threshold</i>	min_mf_distance	max_mf_distance	B_{rule}
Setup1	0.5	2	[1 3 4]	[2.5 7.5 10]	125
Setup2	0.5	0.5	[1 3 4]	[2.5 7.5 10]	125
Setup3	0.5	5	[1 3 4]	[2.5 7.5 10]	125

The initial fuzzy system has 8 rules (initialized to 0) with 2 membership functions in each input dimension as shown in Figs 5.11.

The results of setup 1 are shown in Figures 5.12. It can be seen that the actual output tracks the reference trajectory closely (Fig 5.12a), and thus, the controller successfully controls the position of the magnet. The tracking error is never larger than 0.3, and it quickly converges to near 0 after the set-point changes. The control signal (Fig 5.12c) is always in the desired range $[-5A, 5A]$. The self-structuring activity (Fig 5.12d) occurs in the first 20s of the simulation. Following this, the number of rules remains unchanged at 48. The resulting fuzzy system has membership functions for each input as shown in Figures 5.13.

The self-structuring algorithm suggests that increasing *error_threshold* will result in fewer rules. This is confirmed by setups 2 and 3. Both applications produce the desired performance. Setup 2 results in 64 rules. And setup 3 results in 36 rules

Both examples show that the desired performance can be achieved by different sets of parameters of the self-structuring algorithm. Thus, the choice of parameters is not critical. This gives designers the advantage of freely choosing parameters in practice. This also demonstrates the ability of the self-structuring algorithm to generate a satisfactory structure from different sets of parameters.

5.5. Conclusion

In this chapter, we have proposed a SSDAFC scheme for affine nonlinear systems. The proposed control scheme has some advantages over some existing SSDAFC schemes. It is relatively simpler and more computationally efficient. The maximum number of rules of the fuzzy controller can be set explicitly and thus, never exceed the hardware capacity. The stability is also proved when the structure changes. The use of triangular membership functions increases the interpretability of the rules. Application to an inverted pendulum system and a magnetic levitation system demonstrate the effectiveness of the controller.

It should be noted that the structures generated by the self-structuring algorithms are not the optimal ones. And the main limitation of our approach is that it suffers from “the curse of dimensionality”.

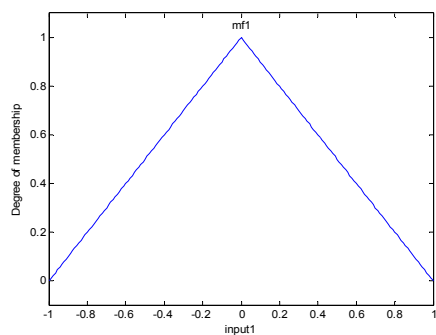


Figure 5.6: initial membership functions for variables x_1 , x_2 , and v

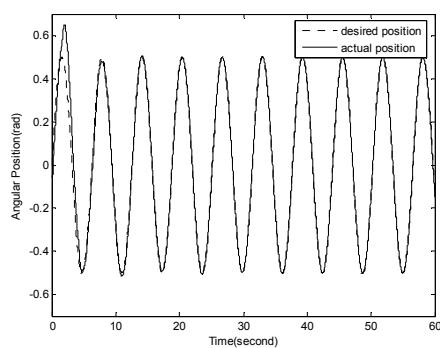


Figure 5.7a: position

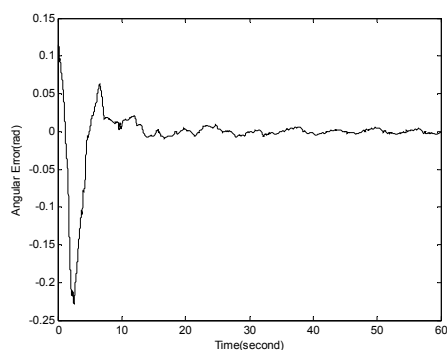


Figure 5.7b: error

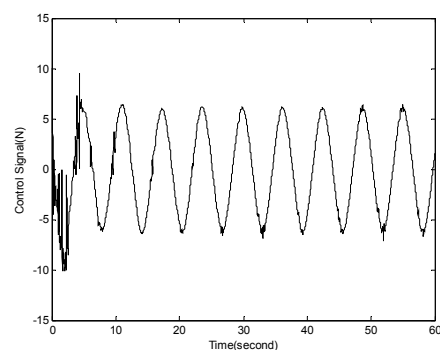


Figure 5.7c: control signal

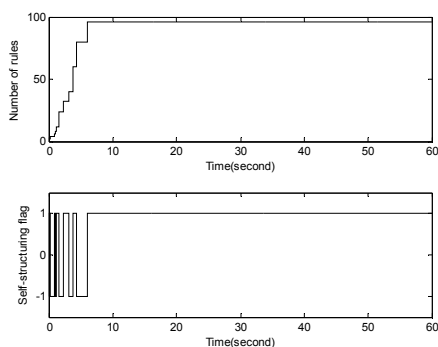


Figure 5.7d: number of rules and self-structuring flag

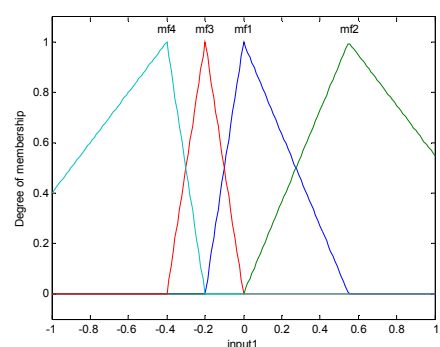


Figure 5.8a: final membership functions for variable x_1

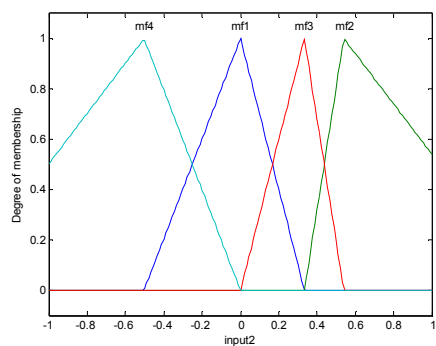


Figure 5.8b: final membership functions for variable x_2

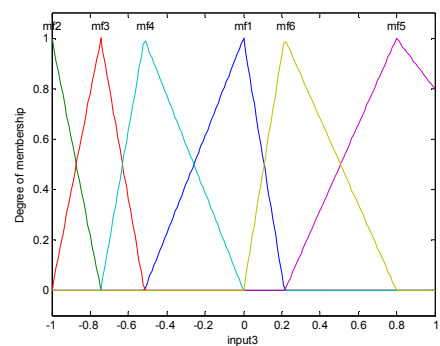


Figure 5.8c: final membership functions for variable v

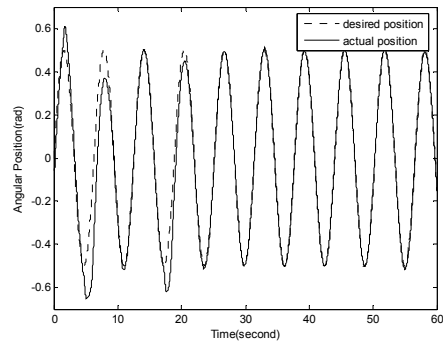


Figure 5.9a: position

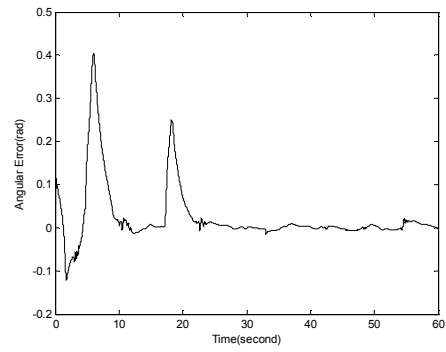


Figure 5.9b: error

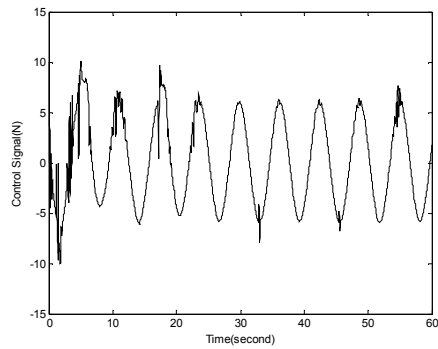


Figure 5.9c: control signal

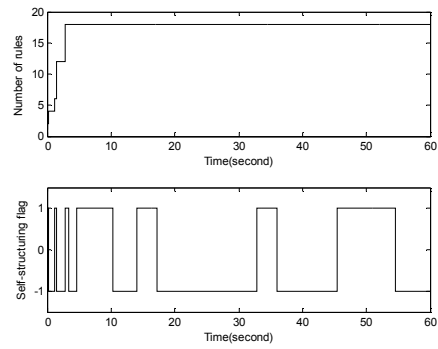
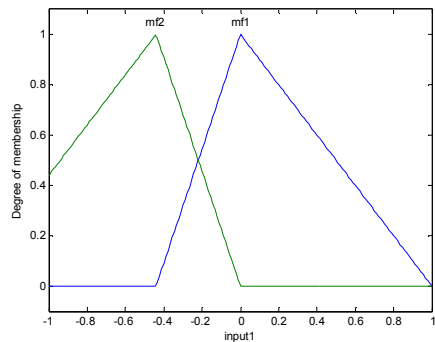
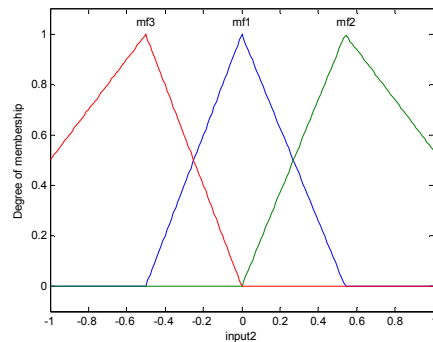
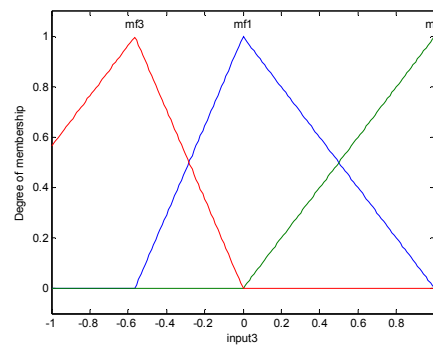
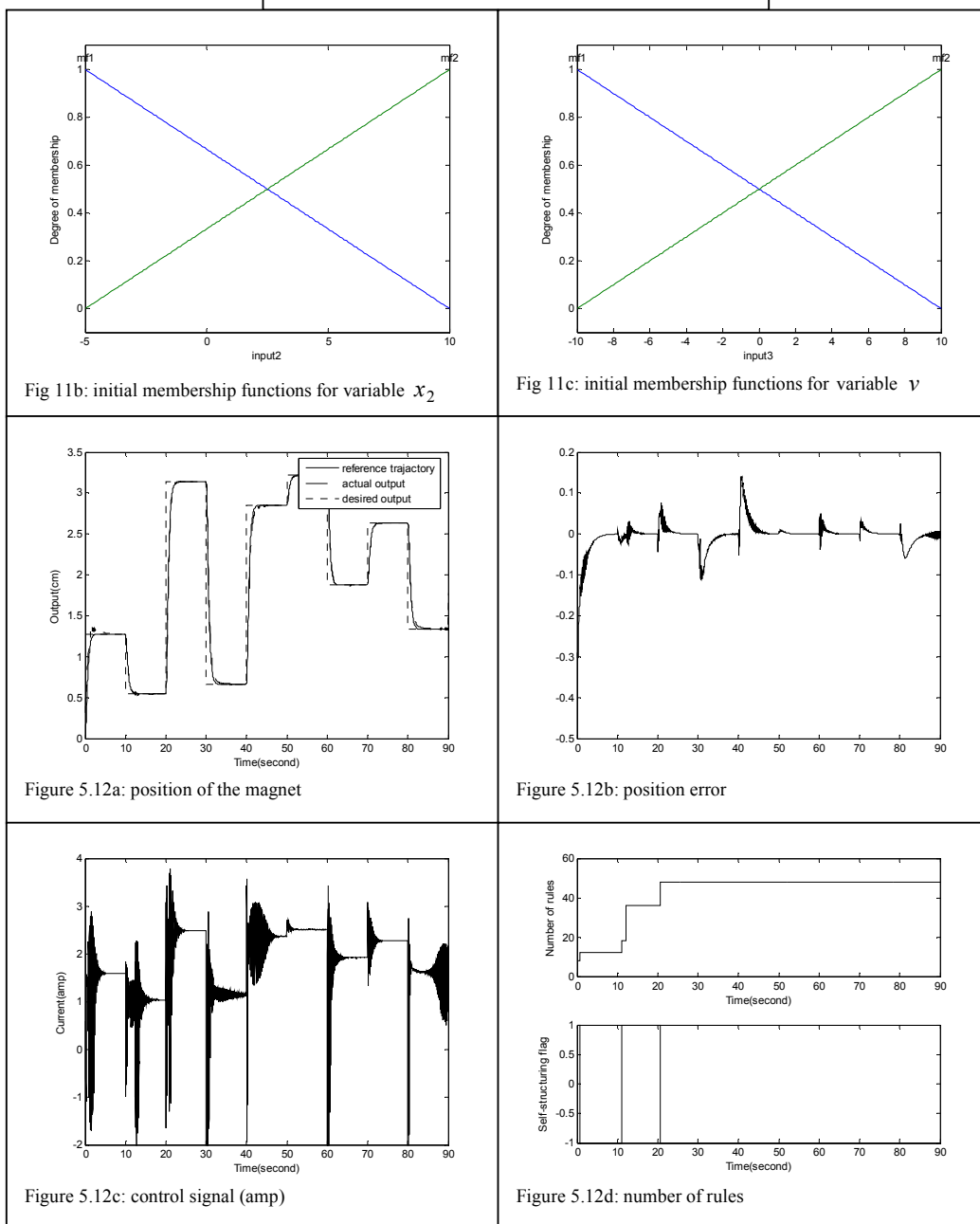
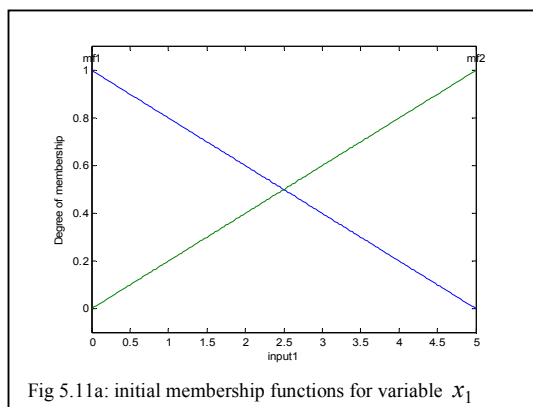
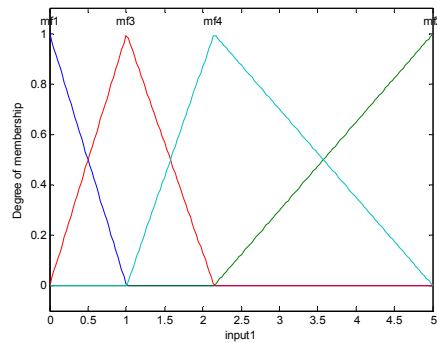
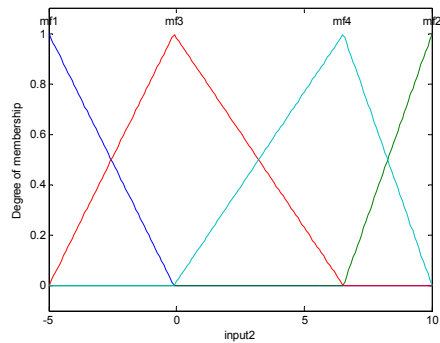
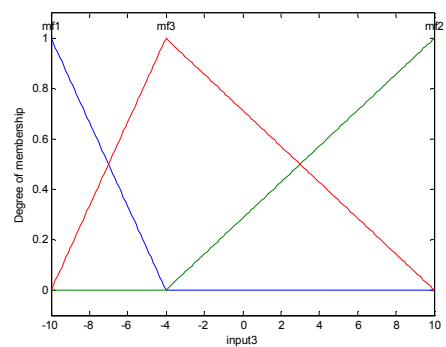


Figure 5.9d: number of rules and self-structuring flag

Figure 5.10a: final membership functions for variable x_1 Figure 5.10b: final membership functions for variable x_2 Figure 5.10c: final membership functions for variable v



Fig 5.13a: final membership functions for variable x_1 Fig 5.13b: final membership functions for variable x_2 Fig 5.13c: final membership functions for variable v

6. Chapter 6

SELF-STRUCTURING DIRECT ADAPTIVE FUZZY CONTROL FOR NON-AFFINE NONLINEAR SYSTEMS³

6.1. Introduction

In chapter 3, 4 and 5, we have discussed AFC of affine nonlinear systems. However, there are many practical nonlinear systems, e.g. chemical reactions and PH neutralization, whose inputs may not be expressed in affine forms. Adaptive intelligent control for non-affine nonlinear systems is more difficult and challenging.

Consider SISO non-affine nonlinear systems described as follows:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= x_3 \\
 &\dots \\
 &\dots \\
 \dot{x}_n &= f(\underline{x}, u) \\
 y &= x_1
 \end{aligned} \tag{6.1}$$

where $u \in R$ is the control input, $y \in R$ is the output, $f(\underline{x}, u)$ is an unknown nonlinear continuous function, $\underline{x} = (x_1, x_2, \dots, x_n)^T$ is the state vector of the system, which is assumed available for measurement. In this chapter, we will investigate SSDAFC of nonlinear systems in the form (6.1).

A short survey about the topic is given in section 6.2. Then, the SSDAFC for nonaffine nonlinear systems (6.1) is given in section 6.3. It is followed by application to two nonaffine nonlinear systems. Finally, conclusion is given in section 6.5

6.2. Literature review

Because the control input does not appear linearly, the well-known feedback linearization technique is not applicable to non-affine nonlinear systems. An explicit expression for the ideal control cannot be obtained. Thus, more complex mathematical tools are needed. In [33], the Taylor series expansion method is used to transform the original system into an affine-like one. In [34], the mean value theorem

³ The content of this chapter has been published in Fuzzy Sets and Systems:

P.A. Phan, and T.J. Gale, "Direct Adaptive Fuzzy Control with a Self-Structuring Algorithm", *Fuzzy Sets and Systems*, in press.

and Nussbaum-Gain functions are used. [2, 35] employ the implicit function theorem and the mean value theorem to show the existence of ideal feedback control. [36-38] propose a pseudo-control scheme and require the contraction condition to show that the pseudo-error can be cancelled by an output of an adaptive neural network (or an adaptive fuzzy system).

Also, more requirements are required. Besides the controllability condition $\frac{\partial(\underline{x}, u)}{\partial u} > 0$, [33] requires the determination of a lower bound g_L such that $\frac{\partial f(\underline{x}, u)}{\partial u} \geq g_L > 0$. [36-38] require the determination of a design parameter c such that $c > \frac{1}{2} \left(\frac{\partial f(\underline{x}, u)}{\partial u} \right)$. [2, 35] require that the derivative of $\frac{\partial f(\underline{x}, u)}{\partial u}$ is bounded and that a design parameter k_v is chosen such as $k_v > k_0$, where k_0 is an unknown positive constant.

To our knowledge, Park et al [37] is the only online self-structuring AFC available for non-affine nonlinear systems. As discussed in chapter 5, the drawbacks of Park's self-structuring algorithm are the even distribution of membership functions and the unrestricted growth of the number of rules. Moreover, in the design of the controller, it is required to select a design parameter c such that $c > \frac{1}{2} \left(\frac{\partial f(\underline{x}, u)}{\partial u} \right)$.

Thus, knowledge of the upper bound of $\frac{\partial f(\underline{x}, u)}{\partial u}$ is needed, or a rather conservative value of c must be chosen. Thus, it is desirable to develop a more efficient AFC scheme for non-affine nonlinear systems.

In this chapter, we propose a new SSDAFC for non-affine nonlinear systems. First, the existence of an implicit ideal control law is shown using the implicit function theorem. Then, using an extension of the universal property, we transform the error dynamic to the same one as for affine nonlinear systems. Thus, theorem 5.1 can be applied. The main contributions are:

- Propose a DAFC scheme for non-affine nonlinear with less restrictions on $\frac{\partial f(\underline{x}, u)}{\partial u}$. The only requirement of the control plant is the controllability condition $\frac{\partial(\underline{x}, u)}{\partial u} > 0$. And there is no restriction on the design parameters.

- Propose using the self-structuring algorithm described in chapter 5 for DAFC of non-affine nonlinear systems.

6.3. SSDAFC for non-affine nonlinear systems

Control objective is to design an adaptive fuzzy controller for non-affine systems described by (6.1) such that:

- The closed-loop system must be stable in the sense that all the variables in the closed-loop system must be bounded.
- The output $y(t)$ of the system follows a continuous reference signal $r(t) \in C^n$.

Assumption 6.1: controllability condition

$$\frac{\partial f(\underline{x}, u)}{\partial u} > 0$$

hold for all $(\underline{x}, u) \in \Omega_{\underline{x}} \times R$ with a controllability region $\Omega_{\underline{x}}$.

Assumption 6.2: Define $\underline{r} = [\dot{r}, \ddot{r}, \ddot{\ddot{r}}, \dots, r^{(n-1)}]^T$. We assume that $\|\underline{r}\| \leq r_0$ and $\|r^{(n)}\| \leq r_1$ with known constants $r_0, r_1 > 0$.

Assumption 6.3 We can determine the upper bound B_{rule} of the required number of rules that achieves the desired approximation accuracy.

6.3.1. Existence of an ideal control law

Let $e = r - y$, $\underline{e} = (e, \dot{e}, \ddot{e}, \dots, e^{(n-1)})^T$, and $\underline{k} = (k_1, k_2, \dots, k_n)^T$ be such that the polynomial $s^n + k_n s^{n-1} + \dots + k_1$ is Hurwitz stable. The ideal control law is chosen to obtain $e^{(n)} = -\underline{k}^T \underline{e} = -k_1 e - k_2 \dot{e} - \dots - k_n e^{(n-1)}$, $e = r - y$, which implies that $\lim_{t \rightarrow +\infty} e = 0$.

$$\text{Let } v = r^{(n)} + \underline{k}^T \underline{e}. \quad (6.2)$$

Adding and subtracting v to (6.1) gives

$$e^{(n)} = -\underline{k}^T \underline{e} - f(\underline{x}, u) - d + v \quad (6.3)$$

Now, we prove that there exists an ideal control signal $u^*(\underline{x}, v)$ such that

$$f(\underline{x}, u^*(\underline{x}, v)) = v \text{ for } (\underline{x}, v) \in R^n \times R.$$

Reintroduce lemma 2.8 that is given in [2]:

Lemma 6.1. Assume that $f : R^n \times R \rightarrow R$ is continuously differentiable and

there exists a positive constant q such that $\left\| \frac{\partial}{\partial u} f(\underline{x}, u) \right\| > q > 0$ for all

$(\underline{x}, u) \in R^n \times R$. Then there exists a unique continuous smooth function $g: R^n \rightarrow R$ such that $f(\underline{x}, g(\underline{x})) = 0$.

Proof is given in [2]◊

Let $\underline{X} = (\underline{x}, v) \in R^{n+1}$ and $F(\underline{X}, u) = f(\underline{x}, u) - v$. We have $\frac{\partial F(\underline{X}, u)}{\partial u} = \frac{\partial f(\underline{x}, u)}{\partial u} - \frac{\partial v}{\partial u} > 0$ as $\frac{\partial v}{\partial u} = 0$ and $\frac{\partial f(\underline{x}, u)}{\partial u} > 0$ (assumption 1). Thus, we can apply lemma 1 for $F(\underline{X}, u)$. Applying lemma 1, there exists a unique continuous smooth function $u^*(\underline{X})$ such that $F(\underline{X}, u^*(\underline{X})) = 0$, i.e.

$$f(\underline{x}, u^*(\underline{X})) = v. \quad (6.4)$$

6.3.2. Stability analysis

From (6.3) and (6.4), we have

$$e^{(n)} = -\underline{k}^T \underline{e} + [f(\underline{x}, u^*(\underline{X})) - f(\underline{x}, u(\underline{X}))] \quad (6.5)$$

Similar to chapter 5, we let $\underline{\theta} = (\underline{\theta}_{ac}^T \quad \underline{\theta}_{in}^T)^T = (\theta_1, \theta_2, \dots, \theta_N)^T$ be the adaptive parameter vector of the final fuzzy controller, in which $\underline{\theta}_{ac} = (\theta_1, \theta_2, \dots, \theta_M)^T$, $(M \leq N)$ is the vector of adaptive parameters already activated, and $\underline{\theta}_{in} = (\theta_{M+1}, \theta_{M+2}, \dots, \theta_N)^T$ is the vector of adaptive parameters not yet generated (inactivated). We will employ a fuzzy system in the form (2.2) to approximate $u^*(\underline{X})$:

$$u(\underline{X}) = \hat{u}(\underline{X} | \underline{\theta}_{ac}) = \sum_{j=1}^M \theta_j \zeta_j(\underline{X}). \quad (6.6)$$

In the literature, the universal approximation property is used to show that there exists a fuzzy controller in the form (6.6) to approximate an ideal control signal with arbitrary accuracy. Here, an extended version of that is introduced:

Lemma 6.2. *Given an arbitrary $\varepsilon^* > 0$, there exist $\underline{\zeta}(\underline{X}) = (\zeta_1(\underline{X}), \zeta_2(\underline{X}), \dots, \zeta_M(\underline{X}))^T$ and an ideal parameter vector $\underline{\theta}^* = (\theta_1, \theta_2, \dots, \theta_M)^T$ such that*

$$f(\underline{x}, u^*(\underline{X})) - f(\underline{x}, u(\underline{X})) = \sum_{j=1}^M c^j (\theta_j^* - \theta_j) \zeta_j(\underline{X}) + \varepsilon \quad (6.7)$$

where $|\varepsilon| \leq \varepsilon^*$ and c^j are some positive constants.

The proof is given in appendix 6.A◊

Substituting (6.7) to (6.5) gives the error dynamics:

$$\underline{e}^{(n)} = -\underline{k}^T \underline{e} + \sum_{j=1}^M c^j (\theta_j^* - \theta_j) \zeta_j(\underline{X}) + \varepsilon. \quad (6.8)$$

In the vector form,

$$\dot{\underline{e}} = \Lambda_C \underline{e} + \underline{b}_C \left[\sum_{j=1}^M c^j (\theta_j^* - \theta_j) \zeta_j(\underline{X}) + \varepsilon \right] \quad (6.9)$$

where

$$\Lambda_C = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -k_1 & -k_2 & -k_3 & \cdots & -k_n \end{pmatrix}, \quad \underline{b}_C = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}.$$

Since Λ_C is a stable matrix, there exists a unique positive definite symmetric $n \times n$ matrix P which satisfies the Lyapunov equation:

$$\Lambda_C^T P + P \Lambda_C = -Q \quad (6.10)$$

where Q is an arbitrary $n \times n$ positive definite matrix chosen such that $\lambda_{\min}(Q) > 1$.

Assumption 6.4 We can determine the upper and lower bounds of the ideal control signal:

$$u_L \leq u^*(\underline{X}) \leq u_U, \quad \forall \underline{X} \in \Omega_{\underline{X}}.$$

As the error dynamic (6.) is the same as equation (4.6), the controllers proposed in chapter 4,5 can be applied to system (6.1). The control scheme is stated in the following theorem

Theorem 6.1 Given system (6.1) satisfying assumptions 6.1, 6.2, 6.3, and 6.4, a controller

$$u = \hat{u}(\underline{X} | \underline{\theta}_{ac}) = \sum_{j=1}^M \theta_j \zeta_j(\underline{X})$$

with the self-structuring algorithm described in section 5.1.1 and the adaptive law

$$\dot{\theta}_j = \begin{cases} \gamma \underline{e}^T P \underline{b}_C \zeta_j(\underline{X}) & \text{if } (u_L < \theta_j < u_U) \\ & \text{or } (\theta_j = u_U \text{ and } \gamma \underline{e}^T P \underline{b}_C \zeta_j(\underline{X}) < 0) \\ & \text{or } (\theta_j = u_L \text{ and } \gamma \underline{e}^T P \underline{b}_C \zeta_j(\underline{X}) > 0) \\ 0 & \text{if } (\theta_j = u_U \text{ and } \gamma \underline{e}^T P \underline{b}_C \zeta_j(\underline{X}) \geq 0) \\ & \text{or } (\theta_j = u_L \text{ and } \gamma \underline{e}^T P \underline{b}_C \zeta_j(\underline{X}) \leq 0) \end{cases}$$

will guarantee that:

vii. The adaptive parameters are bounded:

$$u_L \leq \theta_j \leq u_U, \quad j = 1 \dots M.$$

viii. The tracking error $\underline{e}(t)$ is bounded by:

$$\|\underline{e}(t)\| \leq \sqrt{\frac{2 \max \left\{ V(0), \frac{1}{2\alpha} \left(\frac{c}{\gamma} + \|P\underline{b}_c\|^2 \|\varepsilon^*\|^2 \right) \right\}}{\lambda_{\min}(P)}}, \quad \forall t > 0,$$

in which $\alpha = \frac{(\lambda_{\min}(Q)-1)}{\lambda_{\max}(P)}$, $V(0)$ is a bounded positive constant, and c is a positive

constant that can be made arbitrarily small by tuning the adaptive parameter γ .

ix. The system is Uniformly Ultimately Bounded (UUB), i.e. $\underline{e}(t)$

converges to compact set Ω_ε in finite time:

$$\Omega_\varepsilon = \left\{ \underline{e}(t) \mid \|\underline{e}(t)\| \leq \sqrt{\frac{\|P\underline{b}_c\|^2 \|\varepsilon^*\|^2}{\lambda_{\min}(Q)-1}} \right\}.$$

Proof

As the error dynamic (6.9) is the same as equation (4.6), the proof is the same as in theorem 5.1.

End of poof◊

As the result is the same as in chapter 4 and 5, remarks given in chapters 4 and 5 apply.

6.4. Examples

6.4.1. Application 1

To demonstrate the design procedure and performance, we apply our controller to control a nonaffine nonlinear system that are presented in [2, 35-37]. The dynamic equations of the system are:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_1^2 + 0.15u^3 + 0.1(1+x_2^2)\mu + \sin(0.1u). \\ y &= x_1 \end{aligned}$$

We suppose that there is no prior knowledge of the system except that $\frac{\partial F(\underline{x}, u)}{\partial u} > 0$, which can be easily checked. The initial state is $\underline{x}(0) = [0 \ 0]^T$. The

control objective is to make the output $y(t)$ follow a desired reference $r(t) = \sin(t) + \cos(0.5t)$.

The operating input ranges are chosen as follows:

$$x_1 \in [-5, 5]; x_2 \in [-5, 5]; v \in [-5, 5].$$

The controller's parameters are specified as follows:

$$\underline{k} = [1 \ 1]^T, Q = \begin{bmatrix} 20 & 0 \\ 0 & 10 \end{bmatrix}, P = \begin{bmatrix} 25 & 10 \\ 10 & 15 \end{bmatrix}, \gamma = 100, \sigma = 0.05.$$

The structure-learning parameters are as follows:

$$\varepsilon_0 = 0.5, \text{error_threshold} = 5, \mathbf{min_mf_distance} = [1 \ 1 \ 1],$$

$$\mathbf{max_mf_distance} = [5 \ 5 \ 5], B_{rules} = 30.$$

The initial fuzzy system has only 1 rule (initialized to 0) with 1 membership function in each input dimension as shown in Fig 6.1.

The simulation results are shown in Fig 6.2. It can be observed that the controller successfully controls the nonaffine nonlinear system to track the reference signal $r(t) = \sin(t) + \cos(0.5t)$ (Fig 6.2a). After 10s, the tracking error is within the range $[-0.02, +0.02]$ (Fig 6.2b). The control signal is bounded (Fig 6.2c). The chattering phenomena can be reduced by reducing the adaptive gain γ , but at the expense of increasing the tracking error. Self-structuring happens in the first 5s (Fig 6.2d). A variable called *self-structuring flag* is used to indicate when the self-structuring performs. When the self-structuring flag switches from 1 to -1 or -1 to 1, it indicates a change of the fuzzy system structure has occurred. The final fuzzy system has 12 rules, and the B_{rules} is never reached. The final membership functions in each input dimension are given in Fig 6.3.

To test how the algorithm replaces membership functions and rules, we change the *error_threshold* from 5 to 4, thus more rules are generated. The results are shown in Fig 6.4. The number of rules increases in the first 15s to 24 rules (Fig 6.4d). After that, the number of rules never exceeds B_{rules} . It can be observed that the algorithm replaces membership functions at approximately 19s and 28s. At those moments, there is no degradation in tracking performance (Fig 6.4a). This confirms that replacing membership functions does not effect the performance of the control system. The membership functions of the fuzzy system at $t = 30s$ is given in Fig 6.5.

6.4.2. Application 2

In this application, we control the nonaffine nonlinear system presented in [61, 95]. The system dynamics is:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 + 2x_2 - 2x_1^2 x_2 + \frac{u}{\sqrt{|u|} + 0.1} \\ y &= x_1\end{aligned}$$

It can be seen that the controllability $\frac{\partial F(\underline{x}, u)}{\partial u} > 0$ is satisfied. The initial condition is $\underline{x}(0) = [0.3 \ 0]^T$. The control objective is to make the output $y(t)$ follow a desired reference $r(t) = \frac{\pi}{6} \sin(t)$.

The operating input ranges are chosen as follows:

$$x_1 \in [-2, 2]; \ x_2 \in [-2, 2]; \ v \in [-2, 2]$$

The controller's parameters are specified as follows:

$$\underline{k} = [1 \ 1]^T, \ Q = \begin{bmatrix} 20 & 0 \\ 0 & 10 \end{bmatrix}, \ P = \begin{bmatrix} 25 & 10 \\ 10 & 15 \end{bmatrix}, \ \gamma = 50, \ \sigma = 0.05.$$

The structure-learning parameters are as follows:

$$\begin{aligned}\varepsilon_0 &= 0.5, \ error_threshold = 2, \ \mathbf{min_mf_distance} = [0.4 \ 0.4 \ 0.4], \\ \mathbf{max_mf_distance} &= [2 \ 2 \ 2], \ B_{rules} = 30.\end{aligned}$$

The initial fuzzy system has only 1 rule (initialized to 0) with 1 membership function in each input dimension as shown in Fig 6.6.

The simulation results are shown in Fig 6.7. It can be observed that the controller successfully controls the nonaffine nonlinear system to track the reference signal $r(t) = \frac{\pi}{6} \sin(t)$ (Fig 6.7a). After 8s, the tracking error is within the range $[-0.05, +0.05]$ (Fig 6.7b). The control signal is bounded (Fig 6.7c). The chattering phenomena is not severe in this application. Self-structuring happens in the first 6s (Fig 6.7d). The final fuzzy system has 18 rules, and the B_{rules} is never reached. The final membership functions in each input dimension are given in Fig 6.8.

6.5. Conclusion

In this chapter, we have extended the SSDAFC proposed in chapter 5 to a class of nonaffine nonlinear systems (6.1):

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\dots \\ \dot{x}_n &= f(\underline{x}, u) \\ y &= x_1\end{aligned}$$

All features of the self-structuring algorithm are still valid. Moreover, the control scheme requires less restriction than some existing AFC schemes for nonaffine nonlinear systems. Application to two nonaffine nonlinear systems is shown to demonstrate the approach.

In next chapter, the SSDAFC scheme is further extended to a large class of nonaffine nonlinear systems and a class of nonlinear systems in triangular forms.

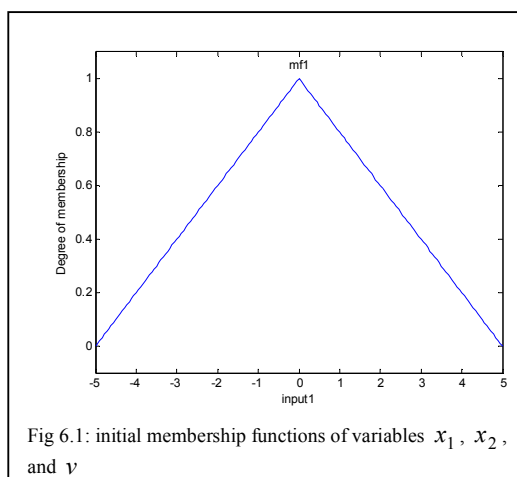


Fig 6.1: initial membership functions of variables x_1 , x_2 , and v

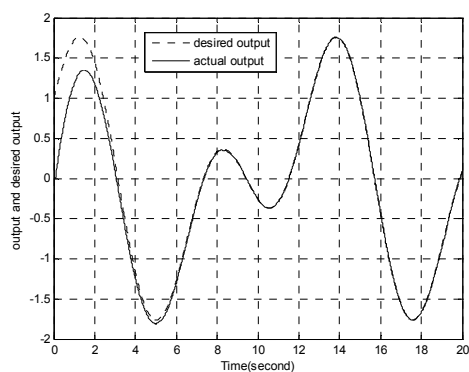


Fig 6.2a: Desired output and actual output

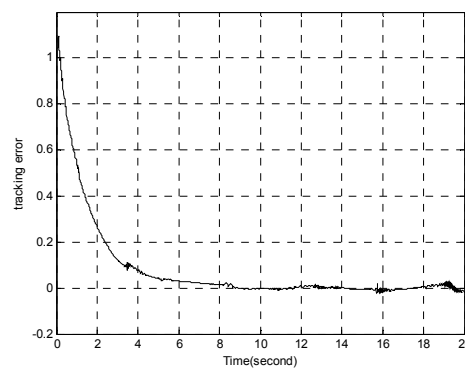


Fig 6.2b: Tracking error

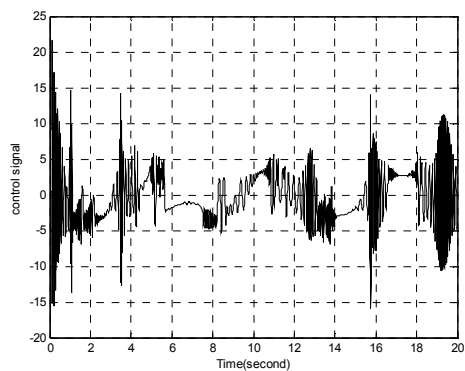


Fig 6.2c: Control signal

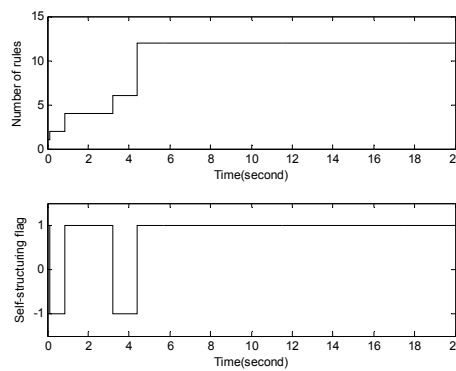
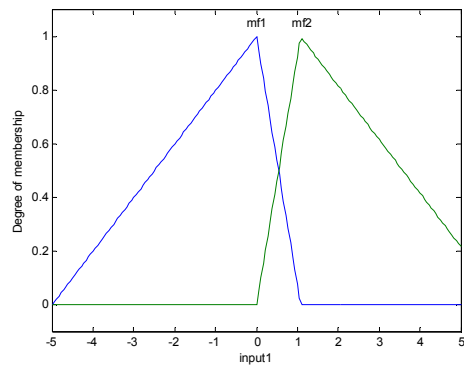
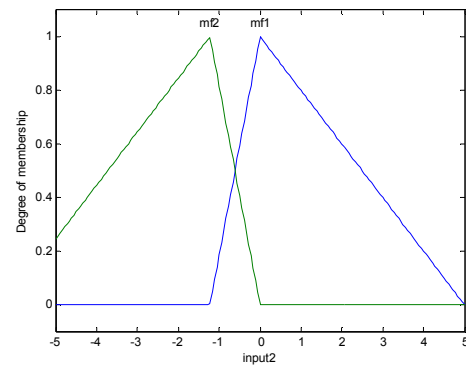
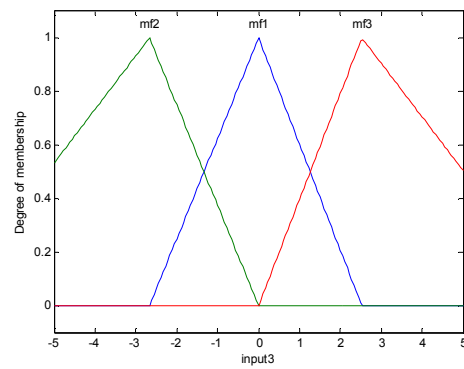


Fig 6.2d: Number of rules and self-structuring flag

Fig 6.3a: final membership functions of variable x_1 Fig 6.3b: final membership functions of variable x_2 Fig 6.3c: final membership functions of variable v

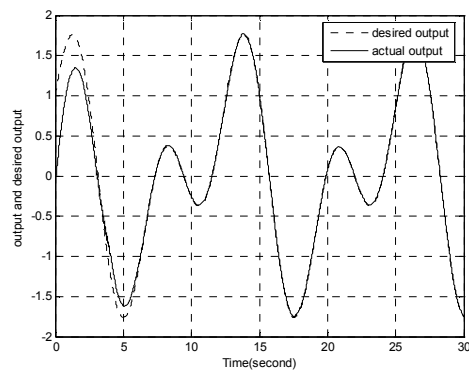


Fig 6.4a: Desired output and actual output for the case $error_threshold = 4$

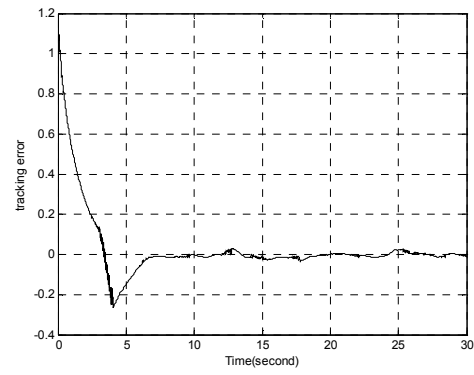


Fig 6.4b: Tracking error for the case $error_threshold = 4$

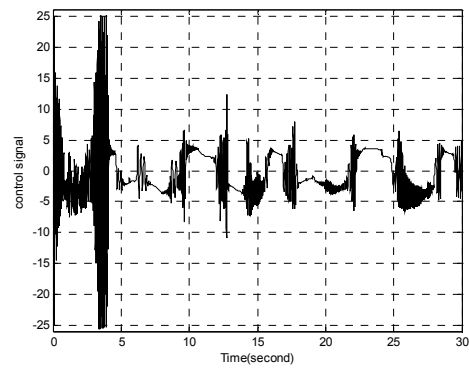


Fig 6.4c: Control signal for the case $error_threshold = 4$

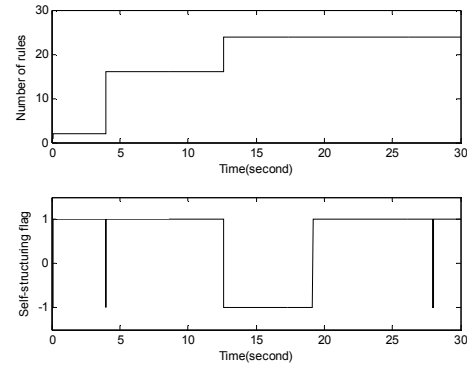


Fig 6.4d: Number of rules and self-structuring flag for the case $error_threshold = 4$

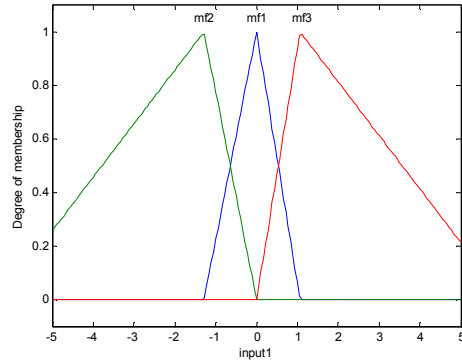


Fig 6.5a: final membership functions of variable x_1 for the case $error_threshold = 4$

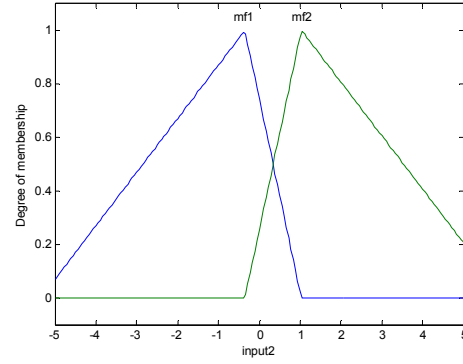


Fig 6.5b: final membership functions of variable x_2 for the case $error_threshold = 4$

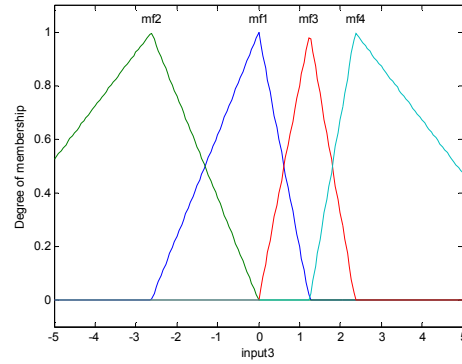


Fig 6.5c: final membership functions of variable v for the case $error_threshold = 4$

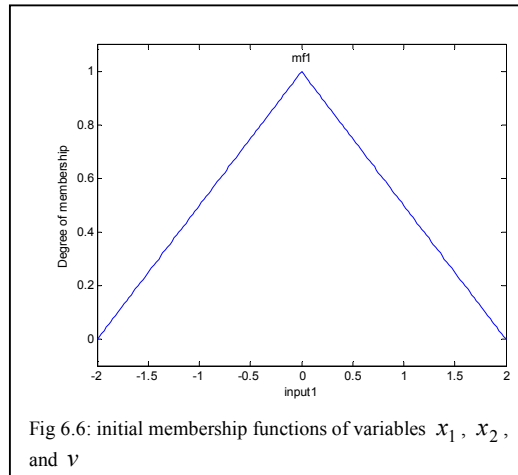


Fig 6.6: initial membership functions of variables x_1 , x_2 , and v

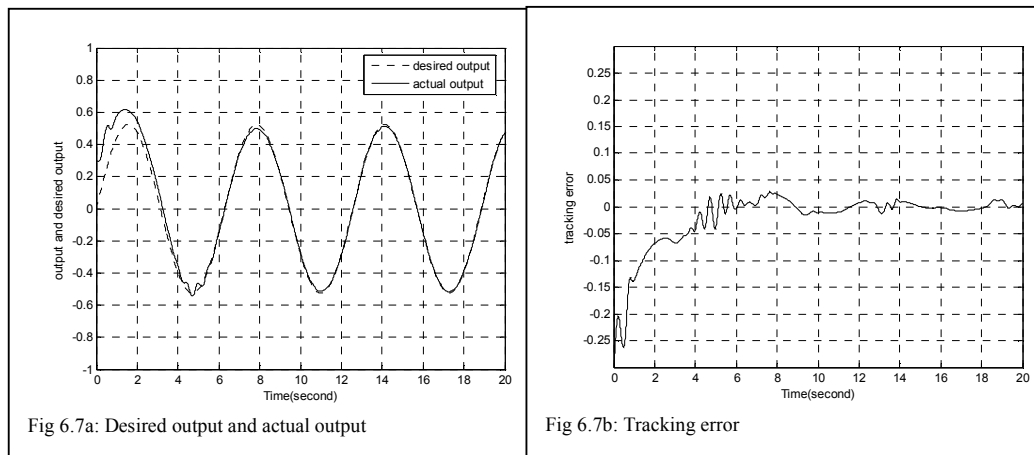


Fig 6.7a: Desired output and actual output

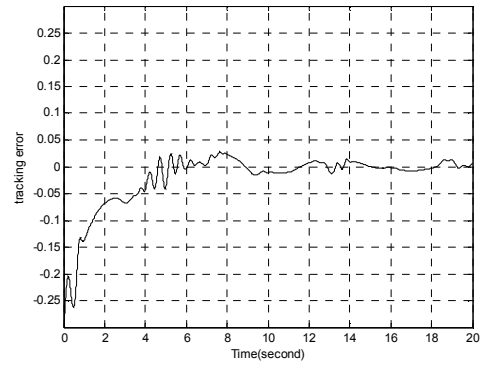


Fig 6.7b: Tracking error

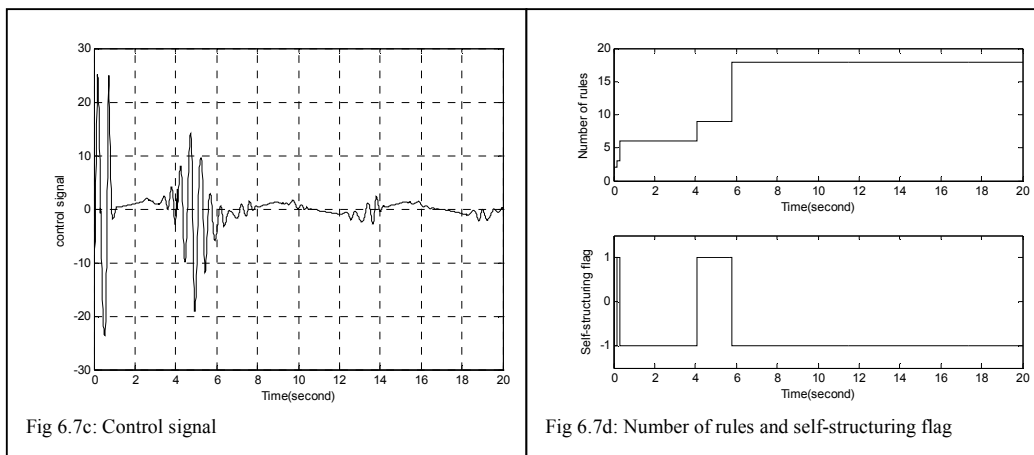


Fig 6.7c: Control signal

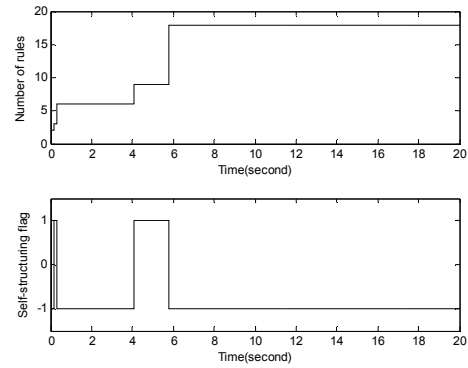
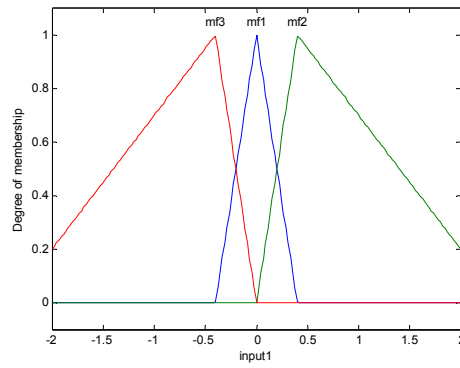
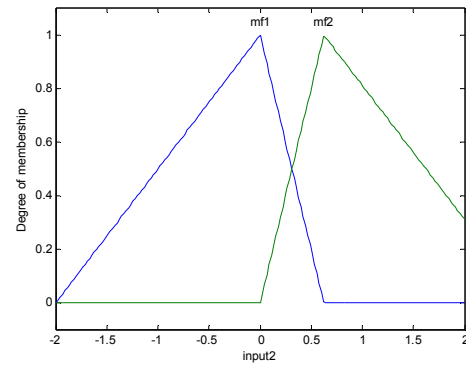
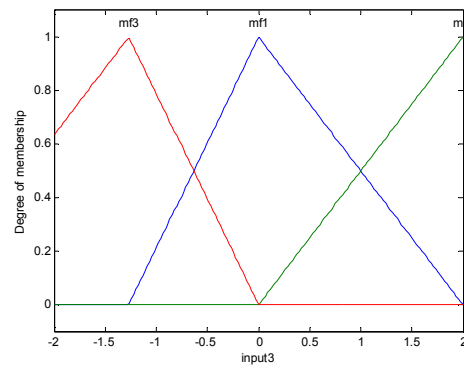


Fig 6.7d: Number of rules and self-structuring flag

Fig 6.8a: final membership functions of variable x_1 Fig 6.8b: final membership functions of variable x_2 Fig 6.8c: final membership functions of variable v

7. Chapter 7

EXTENSION TO THE CONTROL OF OTHER CLASSES OF SISO NON-AFFINE NONLINEAR SYSTEMS

7.1. Introduction

In the last chapter, we have investigated SSDAFC for non-affine systems in the normal form:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= x_3 \\
 &\dots \\
 &\dots \\
 \dot{x}_n &= f(\underline{x}, u) \\
 y &= x_1
 \end{aligned} \tag{7.1}$$

where $u \in R$ is the control input, $y \in R$ is the output, $f(\underline{x}, u)$ is an unknown nonlinear continuous function, $\underline{x} = (x_1, x_2, \dots, x_n)^T$ is the state vector of the system, which is assumed available for measurement. In this chapter, we will extend the results to two broader classes of systems.

In particular, in section 7.2, we will extend SSDAFC to SISO non-affine nonlinear systems in the general form [2, 3, 44, 63, 65, 66]

$$\begin{aligned}
 \underline{\dot{x}} &= \underline{f}(\underline{x}, u) \\
 y &= h(\underline{x})
 \end{aligned} \tag{7.2}$$

where $u \in R$ is the control input, $y \in R$ is the output, $\underline{x} = (x_1, x_2, \dots, x_n)^T$ is the state vector of the system, $\underline{f}(\underline{x}, u) = (f_1(\underline{x}, u), f_2(\underline{x}, u), \dots, f_n(\underline{x}, u))^T$ is a vector of unknown nonlinear continuous functions, and $h(\underline{x})$ is an unknown continuous function.

Then, in section 7.3, we will consider the control of systems in triangular form [2, 34, 56]:

$$\begin{aligned}
 \dot{x}_1 &= f_1(x_1, \dots, x_{i+1}) \quad i = 1, \dots, n-1 \\
 \dot{x}_n &= f_n(x_1, \dots, x_n, u) \\
 y &= x_1
 \end{aligned} \tag{7.3}$$

where $u \in R$ is the control input, $y \in R$ is the output, $\underline{x} = (x_1, x_2, \dots, x_n)^T$ is the state vector of the system, $f_i(\bullet)$, $i = 1 \dots n$ are unknown continuous functions.

As we will show later, the control of systems (7.2) and (7.3) requires knowledge of the output y and its derivatives $y^{(i)}$, $i = 1 \dots k$, $k \leq n$. In practice, the derivatives of y are sometimes not available for measurement. Output feedback control, in which only the output is required, is an approach to overcome this difficulty. In section 7.4, we will use a high-gain observer to observe the derivatives of y and thus propose an output feedback SSDAFC scheme.

7.2. SSDAFC of systems in the form (7.2)

Fixed-structured adaptive intelligent control has been proposed for system (7.2) in the literature [2, 3, 44, 63, 65, 66]. The key idea is to show that system (7.2) can be transformed to the form (7.1) if certain conditions are satisfied. Then, adaptive intelligent control for system (7.1) can be applied to system (7.2).

However, to our knowledge, no self-structuring adaptive intelligent control approach for system (7.2) has been in the literature. Therefore, inspired by the previous works, we first show that system (7.2) can be transformed to the form (7.1). Then, our SSDAFC proposed in chapter 6 can be applied to system (7.2).

Definition 7.1 Lie derivative

Let $L_f h$ denote the Lie derivative of the function $h(\underline{x})$ with respect to the vector field $f(\underline{x}, u)$:

$$L_f h = \frac{\partial[h(\underline{x})]}{\partial \underline{x}} f(\underline{x}, u)$$

Higher-order derivatives are defined recursively as $L_f^k h = L_f(L_f^{k-1} h)$, $k > 1$.

Definition 7.2 Strong relative degree

For $\underline{x} \in \Omega_{\underline{x}} \subset R^n$ and $u \in \Omega_u \subset R$, system (7.2) is said to have a strong relative degree ρ in $\Omega_{\underline{x}} \times \Omega_u$ if there exists a positive integer $1 \leq \rho \leq \infty$ such that

$$\frac{\partial[L_f^i h]}{\partial u} = 0, \quad i = 0, 1, \dots, \rho - 1, \quad \frac{\partial[L_f^\rho h]}{\partial u} \neq 0$$

for all $(\underline{x}, u) \in \Omega_{\underline{x}} \times \Omega_u$.

To continue, we consider two cases: system (7.2) with strong relative degree $\rho = n$, and system (7.2) with strong relative degree $\rho < n$.

7.2.1. Control of system (7.2) with strong relative degree $\rho = n$

With $\rho = n$, system (7.2) can be transformed into a normal form (7.1) [2]

$$\begin{cases} \dot{\xi}_i = \xi_{i+1}, i = 1, \dots, n-1 \\ \dot{\xi}_n = b(\underline{\xi}, u) \\ y = \xi_1 \end{cases} \quad (7.4)$$

where $\xi_i = L_f^{i-1} h(\underline{x})$, $i = 1, \dots, n$, $b(\underline{\xi}, u) = L_f^n h(\underline{x})$, and $\underline{\xi} = [\xi_1 \ \xi_2 \ \dots \ \xi_n]^T \in \Omega_\xi$, $\Omega_\xi = \{\underline{\xi} | \underline{x} \in \Omega_x\}$.

From definition 7.2, $\frac{\partial b(\underline{\xi}, u)}{\partial u} \neq 0 \quad \forall \underline{\xi} \in \Omega_\xi, u \in \Omega_u$, i.e. assumption 6.1 is

satisfied. Thus, given system (7.2) with strong relative degree $\rho = n$, the self-structuring DAFC proposed in 6.3 applied to system (7.4) guarantees that all signal of system (7.2) are bounded and the tracking error is uniformly ultimately bounded (UUB).

Remark 7.1 After the transformation, the output and its derivatives $(y, \dot{y}, \dots, y^{(n-1)})^T = (\xi_1, \xi_2, \dots, \xi_{n-1})^T$ are needed to construct the controller. If these signals are not available, they need to be estimated. In section 7.4, we will present an output feedback SSDAFC scheme, in which only the output is measurable. Its derivatives will be estimated using observers.

7.2.2. Control of system (7.2) with strong relative degree $\rho < n$

With $\rho < n$, after the transformation, we have the system in the normal form:

$$\begin{cases} \dot{\xi}_i = \xi_{i+1}, i = 1, \dots, \rho-1 \\ \dot{\xi}_\rho = b(\underline{\xi}, \underline{\eta}, u) \\ \dot{\underline{\eta}} = \underline{q}(\underline{\xi}, \underline{\eta}, u) \\ y = \xi_1 \end{cases} \quad (7.5)$$

where $\xi_i = L_f^{i-1} h(\underline{x})$, $i = 1, \dots, \rho$, $b(\underline{\xi}, u) = L_f^\rho h(\underline{x})$, $\underline{q}_i(\underline{\xi}, \underline{\eta}, u) = L_f^{i+\rho} h(\underline{x})$, $i = 1, \dots, n-\rho$, $(\underline{\xi}, \underline{\eta}) \in \Omega_\xi \times \Omega_\eta$, $\Omega_\xi \times \Omega_\eta = \{\underline{\xi}, \underline{\eta} | \underline{x} \in \Omega_x\}$.

If applying the SSDAFC scheme in section 6.3 to the ξ -subsystem

$$\begin{cases} \dot{\xi}_i = \xi_{i+1}, i = 1, \dots, \rho - 1 \\ \dot{\xi}_\rho = b(\underline{\xi}, \underline{\eta}, u) \\ y = \xi_1 \end{cases}, \quad (7.6)$$

the states $\underline{\eta}$ is completely unobservable. Thus, $\underline{\eta}$ is not guaranteed to be bounded. The dynamics $\dot{\underline{\eta}} = \underline{q}(\underline{\xi}, \underline{\eta}, u)$ is called **the internal dynamics**. And with $\underline{\xi} = 0$, $\dot{\underline{\eta}} = \underline{q}(0, \underline{\eta}, u)$ is addressed as **the zero dynamics**.

To assure the boundedness of the internal dynamics, the following assumption is required [63, 65, 66]:

Assumption 7.1 The system (7.2) is hyperbolically minimum-phase, i.e. the zero dynamics is exponentially stable.

In [63, 65, 66], the authors show that assumption 7.1 implies that bounded $\underline{\xi}$ leads to bounded $\underline{\eta}$.

Therefore, given system (7.2) with zero dynamics, i.e. with strong relative degree $\rho < n$, if assumption 7.1 is satisfied, the self-structuring DAFC scheme proposed in section 6.2 applied to the ξ -subsystem (7.6) guarantees all signals of (7.2) are bounded and the tracking error is UUB.

7.3. SSDAFC of systems in the triangular form (7.3)

The class of systems in the triangular form is a very popular class of SISO nonlinear systems. This class includes both strict-feedback systems and pure-feedback systems. A brief review of adaptive intelligent control of systems in the triangular form (7.3) has been given in chapter 2. Adaptive back-stepping is the main technique. Using the adaptive back-stepping technique, we can construct a backstepping-based SSDAFC for system (7.3), in which an adaptive fuzzy system equipped with the self-structuring algorithm proposed in chapter 5 is used at every step to approximate the virtual control at every step. However, a serious draw back of the backstepping technique is that it needs at least one adaptive intelligent system at every step. This dramatically increases the complexity of the controller as the order of the system increases.

Therefore, here, we propose a SSDAFC for system (7.3), in which only one adaptive fuzzy system is required no matter what the order of the system is. The idea

is to show that we can transform system (7.3) to the form (7.1). Then, the SSDAFC proposed in chapter 6 can applied to system (7.3).

First, we need to state the controllability condition for system (7.3) as commonly made in the literature:

Assumption 7.2 System (7.3) satisfies:

$$\frac{\partial f_i(x_1, \dots, x_{i+1})}{\partial x_{i+1}} \neq 0, \quad i=1, \dots, n-1 \quad \text{and} \quad \frac{\partial f_n(x_1, \dots, x_n, u)}{\partial u} \neq 0 \quad \forall \underline{x} \in \Omega_x \subset R^n. \quad (7.7)$$

Now, we need to show that if assumption 7.2 is satisfied then system (7.3) has strong relative degree $\rho = n$:

- The first Lie derivative of the output y of system (7.3) is:

$$L_f y = \frac{\partial [y(\underline{x})]}{\partial \underline{x}} \underline{f}(\underline{x}, u) = \frac{\partial x_1}{\partial x_1} f_1(x_1, x_2) = f_1(x_1, x_2).$$

$$\text{Obviously, } \frac{\partial [L_f y]}{\partial u} = 0 \quad \forall \underline{x} \in \Omega_x \subset R^n.$$

- The second Lie derivative of the output y of system (7.3) is:

$$L_f^2 y = \frac{\partial [L_f y]}{\partial \underline{x}} \underline{f}(\underline{x}, u) = \frac{\partial f_1(x_1, x_2)}{\partial x_1} f_1(x_1, x_2) + \frac{\partial f_1(x_1, x_2)}{\partial x_2} f_2(x_1, x_2, x_3).$$

$$\text{From assumption 7.2, } \frac{\partial f_1(x_1, x_2)}{\partial x_2} \neq 0 \quad \text{and} \quad \frac{\partial f_2(x_1, x_2, x_3)}{\partial x_3} \neq 0 \quad \forall \underline{x} \in \Omega_x \subset R^n,$$

the right-hand side is guaranteed to depend on x_3 . Thus, we can let

$$\frac{\partial f_1(x_1, x_2)}{\partial x_1} f_1(x_1, x_2) + \frac{\partial f_1(x_1, x_2)}{\partial x_2} f_2(x_1, x_2, x_3) = F_2(x_1, x_2, x_3) \quad \text{in} \quad \text{which}$$

$$\frac{\partial F_2(x_1, x_2, x_3)}{\partial x_3} \neq 0, \quad \forall \underline{x} \in \Omega_x \subset R^n.$$

$$\text{Obviously, } \frac{\partial [L_f^2 y]}{\partial u} = \frac{\partial F_2(x_1, x_2, x_3)}{\partial u} = 0 \quad \forall \underline{x} \in \Omega_x \subset R^n.$$

- The third Lie derivative of the output y of system (7.3) is:

$$L_f^3 y = \frac{\partial [L_f^2 y]}{\partial \underline{x}} \underline{f}(\underline{x}, u) = \sum_{k=1}^2 \frac{\partial F_2(x_1, x_2, x_3)}{\partial x_k} f_k(x_1, \dots, x_{k+1}) + \frac{\partial F_2(x_1, x_2, x_3)}{\partial x_3} f_3(x_1, \dots, x_4).$$

$$\text{Since } \frac{\partial F_2(x_1, x_2, x_3)}{\partial x_3} \neq 0 \quad \text{and} \quad \frac{\partial f_3(x_1, x_2, x_3, x_4)}{\partial x_4} \neq 0 \quad \forall \underline{x} \in \Omega_x \subset R^n, \quad \text{the right-}$$

hand side is guaranteed to depend on x_4 . Thus, we can let

$$\sum_{k=1}^2 \frac{\partial F_2(x_1, x_2, x_3)}{x_k} f_k(x_1, \dots, x_{k+1}) + \frac{\partial F_2(x_1, x_2, x_3)}{\partial x_3} f_3(x_1, \dots, x_4) = F_3(x_1, x_2, x_3, x_4) \quad \text{in}$$

which $\frac{\partial F_3(x_1, x_2, x_3, x_4)}{\partial x_4} \neq 0 \quad \forall \underline{x} \in \Omega_x \subset R^n$.

Obviously, $\frac{\partial [L_f^3 y]}{\partial u} = \frac{\partial F_3(x_1, x_2, x_3, x_4)}{\partial u} = 0 \quad \forall \underline{x} \in \Omega_x \subset R^n$.

- The i -th Lie derivative ($i = 4 \dots n-1$) of the output y of system (7.3) is:

$$L_f^i y = \frac{\partial [L_f^{i-1} y]}{\partial \underline{x}} \underline{f}(\underline{x}, u) = \sum_{k=1}^{i-1} \frac{\partial F_{i-1}(x_1, \dots, x_i)}{x_k} f_k(x_1, \dots, x_{k+1}) + \frac{\partial F_{i-1}(x_1, \dots, x_i)}{\partial x_i} f_i(x_1, \dots, x_{i+1})$$

Since $\frac{\partial F_{i-1}(x_1, \dots, x_i)}{\partial x_i} \neq 0$ and $\frac{\partial f_i(x_1, \dots, x_{i+1})}{\partial x_{i+1}} \neq 0 \quad \forall \underline{x} \in \Omega_x \subset R^n$, the right-hand

side is guaranteed to depend on x_{i+1} . Thus, we can let

$$\sum_{k=1}^{i-1} \frac{\partial F_{i-1}(x_1, \dots, x_i)}{x_k} f_k(x_1, \dots, x_{k+1}) + \frac{\partial F_{i-1}(x_1, \dots, x_i)}{\partial x_i} f_i(x_1, \dots, x_{i+1}) = F_i(x_1, \dots, x_{i+1}) \quad \text{in}$$

which $\frac{\partial F_i(x_1, \dots, x_{i+1})}{\partial x_{i+1}} \neq 0 \quad \forall \underline{x} \in \Omega_x \subset R^n$.

Obviously, $\frac{\partial [L_f^i y]}{\partial u} = \frac{\partial F_i(x_1, \dots, x_{i+1})}{\partial u} = 0 \quad \forall \underline{x} \in \Omega_x \subset R^n$.

- The n -th Lie derivative of the output y of system (7.3) is:

$$L_f^n y = \frac{\partial [L_f^{n-1} y]}{\partial \underline{x}} \underline{f}(\underline{x}, u) = \sum_{k=1}^{n-1} \frac{\partial F_{n-1}(x_1, \dots, x_n)}{x_k} f_k(x_1, \dots, x_{k+1}) + \frac{\partial F_{n-1}(x_1, \dots, x_n)}{\partial x_n} f_n(x_1, \dots, x_n, u)$$

Since $\frac{\partial F_{n-1}(x_1, \dots, x_n)}{\partial x_n} \neq 0$ and $\frac{\partial f_n(x_1, \dots, x_n, u)}{\partial u} \neq 0 \quad \forall \underline{x} \in \Omega_x \subset R^n$, the right-

hand side is guaranteed to depend on u . Thus, we can let

$$\sum_{k=1}^{n-1} \frac{\partial F_{n-1}(x_1, \dots, x_n)}{x_k} f_k(x_1, \dots, x_{k+1}) + \frac{\partial F_{n-1}(x_1, \dots, x_n)}{\partial x_n} f_n(x_1, \dots, x_n, u) = F_n(x_1, \dots, x_n, u)$$

in which $\frac{\partial F_n(x_1, \dots, x_n, u)}{\partial u} \neq 0 \quad \forall \underline{x} \in \Omega \subset R^n$. Therefore,

$$\frac{\partial [L_f^n y]}{\partial u} = \frac{\partial F_n(x_1, \dots, x_n, u)}{\partial u} \neq 0 \quad \forall \underline{x} \in \Omega_x \subset R^n.$$

Thus, from definition 7.2, we conclude that system (7.3) has strong relative degree $\rho = n \quad \forall \underline{x} \in \Omega_x \subset R^n$.

As system (7.3) belongs to class (7.2) and assumption 7.2 implies strong relative degree $\rho = n$, system (7.3) can be transformed to the form (7.4). Thus, the SSDAFC proposed in 6.3 applied to system (7.4) guarantees that all signals of system (7.3) are bounded and the tracking error is UUB.

7.4. Output feedback SSDAFC

As shown in section 7.2 and 7.3, control of systems (7.2) and (7.3) requires the output and its derivatives $(y, \dot{y}, \dots, y^{(\rho-1)})^T = (\xi_1, \xi_2, \dots, \xi_{\rho-1})^T$ where ρ is the strong relative degree ($\rho \leq n$). This section deals with the case where only the output is available for measurement.

Output feedback adaptive intelligent control has been proposed in the literature. Observers are the main tool to estimate the unavailable signals. [44, 58] propose using high gain observers to estimate the required derivatives of the outputs. [61, 62] propose using linear observers to observe the error dynamics. However, in [62], the role of the fuzzy-neural controller is undermined as the nonlinearity of the system is compensated by a high gain robust control term. One non-observer approach is proposed in [38], in which linear dynamic compensators and low-pass filters are used to generate the adaptive signal, and input/output history are used as inputs to NNs instead of the derivatives of the system output.

In this section, we employ a high-gain observer to estimate the derivatives of the output. The main advantage of using high-gain observers is the design of observers is separate from the design of adaptive intelligent controllers. Thus, the design can be divided into 2 steps. First, a SSDAFC is designed assuming all signals are available. Then, a high-gain observer is designed to observe the unmeasurable derivatives.

The high-gain observer presented in [2] is given as follows:

Lemma 7.1. *Suppose the system output $y(t)$ and its first n derivatives are bounded, so that $|y^{(k)}| < Y_k$ with positive constants Y_k . Consider the following linear system*

$$\begin{cases} \exists \dot{\pi}_i = \pi_{i+1}, i = 1, \dots, n-1 \\ \exists \dot{\pi}_n = -\bar{\lambda}_1 \pi_n - \bar{\lambda}_2 \pi_{n-1} - \dots - \bar{\lambda}_{n-1} \pi_2 - \pi_1 + y(t) \end{cases} \quad (7.7)$$

where ϑ is any small positive constant and the parameters $\bar{\lambda}_1$ to $\bar{\lambda}_{n-1}$ are chosen such that the polynomial $s^n + \bar{\lambda}_1 s^{n-1} + \dots + \bar{\lambda}_{n-1} s + 1$ is Hurwitz stable. Then

$$(i) \quad \frac{\pi_{k+1}}{\vartheta^k} - y^{(k)} = -\vartheta \psi^{(k+1)}, k=1, \dots, n-1$$

$$\text{where } \psi = \pi_n + \bar{\lambda}_1 \pi_{n+1} + \dots + \bar{\lambda}_{n-1} \pi_1.$$

(ii) There exist positive constants t^* and h_k only depending on Y_k , and $\bar{\lambda}_i$, $i=1, \dots, n-1$ such that for all $t > t^*$ we have $|\psi^{(k)}| \leq h_k$, $k=2, 3, \dots, n$.

Proof: The proof is given here for completeness. From (7.7), we have:

$$\frac{\pi_2}{\vartheta} - \dot{y} = \frac{\pi_2}{\vartheta} - \vartheta \ddot{\pi}_n - \bar{\lambda}_1 \dot{\pi}_{n-1} - \bar{\lambda}_2 \dot{\pi}_{n-2} - \dots - \bar{\lambda}_{n-1} \dot{\pi}_2 - \dot{\pi}_1$$

From (7.7) and the above equation yields

$$\frac{\pi_2}{\vartheta} - \dot{y} = -\vartheta \ddot{\psi}$$

By differentiating the above equation and utilizing (7.7), item (i) follows.

The derivatives of the vector $\underline{\pi} = [\pi_1 \quad \pi_2 \quad \dots \quad \pi_n]^T$ may be computed as follows:

$$\begin{aligned} \underline{\pi}^{(j)}(t) = & \frac{1}{\vartheta^j} A^j \exp\left(\frac{At}{\vartheta}\right) \left[\underline{\pi}(0) + A^{-1} \underline{b} y(0) + \dots + \vartheta^{j-1} A^{-j} \underline{b} y^{(j-1)}(0) \right] \\ & + \frac{1}{\vartheta} \exp\left(\frac{At}{\vartheta}\right) \int_0^t \exp\left(-\frac{A\tau}{\vartheta}\right) \underline{b} y^{(j)}(\tau) d\tau, j=1, 2, \dots, n \end{aligned} \quad (7.8)$$

where A is the matrix corresponding to the homogeneous part of (7.7), and independent of ϑ , and $\underline{b} = [0 \quad 0 \quad \dots \quad 1]^T$. Since $\underline{\xi}$ belongs to the compact set Ω_ξ and u is bounded, there exist constants $Y_j > 0$ such that $|y^{(j)}| \leq Y_j$. Then, for any $\delta > 0$, we may find a constant $t^* > 0$ such that, for all $t > t^*$, the first term

$$\frac{1}{\vartheta^j} A^j \exp\left(\frac{At}{\vartheta}\right) \left[\underline{\pi}(0) + A^{-1} \underline{b} y(0) + \dots + \vartheta^{j-1} A^{-j} \underline{b} y^{(j-1)}(0) \right]$$

in (7.8) is bounded by δY_j for each j . Further, since $|y^{(j)}| < Y_j$, there exist constants D_j , which is independent of ϑ , such that, for each j , the second term in (7.8)

$$\frac{1}{\vartheta} \exp\left(\frac{At}{\vartheta}\right) \int_0^t \exp\left(-\frac{A\tau}{\vartheta}\right) \underline{b} y^{(j)}(\tau) d\tau < D_j Y_j$$

Now, fix an arbitrarily small δ^* . Then for $t > t^*$, we have $|\psi^{(j)}| \leq h_j$ where $h_j = B(D_j + \delta^*)Y_j$ with B the norm of the vector $[1 \quad \bar{\lambda}_1 \quad \dots \quad \bar{\lambda}_{n-1}]$. As D_j , δ^* , Y_j , and B are independent of ϑ , the proof is completed. \diamond

It should be noted that lemma 7.1 also holds for $k = n$. We need this later to show the stability of the controller.

The output SSDAFC is proposed as follows:

Theorem 7.1 *Given system (7.1) with only the output y measurable, if the state variables are estimated as*

$$\begin{aligned} \hat{x}_1 &= \pi_1 \\ \hat{x}_2 &= \frac{\pi_2}{\vartheta} \\ \hat{x}_3 &= \frac{\pi_3}{\vartheta^2} \\ &\dots \\ \hat{x}_n &= \frac{\pi_n}{\vartheta^{n-1}} \end{aligned} \quad (7.9)$$

where $\underline{\pi} = (\pi_1 \quad \pi_2 \quad \dots \quad \pi_n)^T$ is estimated using the observer (7.7), then a controller

$$u = \hat{u}(\hat{X}|\underline{\theta}_{ac}) = \sum_{j=1}^M \theta_j \zeta_j(\hat{X}) \quad (7.10)$$

with the self-structuring algorithm described in section 5.1.1 and the adaptive law

$$\dot{\theta}_j = \begin{cases} \gamma \hat{e}^T P \underline{b}_c \zeta_j(\hat{X}) & \text{if } (u_L < \theta_j < u_U) \\ & \text{or } (\theta_j = u_U \text{ and } \gamma \hat{e}^T P \underline{b}_c \zeta_j(\hat{X}) < 0) \\ & \text{or } (\theta_j = u_L \text{ and } \gamma \hat{e}^T P \underline{b}_c \zeta_j(\hat{X}) > 0) \\ 0 & \text{if } (\theta_j = u_U \text{ and } \gamma \hat{e}^T P \underline{b}_c \zeta_j(\hat{X}) \geq 0) \\ & \text{or } (\theta_j = u_L \text{ and } \gamma \hat{e}^T P \underline{b}_c \zeta_j(\hat{X}) \leq 0) \end{cases} \quad (7.11)$$

in which $\hat{X} = (\hat{x}, \hat{v})$ and $\hat{e} = \underline{r} - \hat{x}$, will guarantee that

$$i. \quad \|\underline{e}(t)\| \leq \sqrt{\frac{2 \max \left\{ V(0), \frac{1}{2\alpha} \left(\frac{c}{\gamma} + \|P \underline{b}_c\|^2 \|\varepsilon^* + \vartheta H\|^2 \right) \right\}}{\lambda_{\min}(P)}} + \vartheta \|h\|, \quad \forall t > 0,$$

ii. and $\underline{e}(t)$ converges to compact set

$$\Omega_{\underline{e}} = \left\{ \underline{e}(t) \left| \|\underline{e}(t)\| \leq \sqrt{\frac{\|P\underline{b}_c\|^2 \|\varepsilon^* + \vartheta H\|^2}{\lambda_{\min}(Q) - 1}} + \vartheta \|\underline{h}\| \right. \right\}.$$

Proof

The proof includes 3 steps. The first step is to derive the dynamics $\dot{\hat{\underline{x}}}$ of the observed states $\hat{\underline{x}}$. Then, we show that a SSDAFC applied to the system of the observed states $\hat{\underline{x}}$ guarantees that the observed tracking error $\hat{\underline{e}} = \underline{r} - \hat{\underline{x}}$ is UUB. Finally, as $\underline{x} - \hat{\underline{x}}$ is bounded, the actual tracking error $\underline{e} = \underline{r} - \underline{x}$ is also UUB.

• Step 1:

From lemma 7.1(i),

$$\frac{\pi_n}{\vartheta^{n-1}} - y^{(n-1)} = \hat{x}_n - y^{(n-1)} = -\vartheta \psi^{(n)}.$$

Differentiate it, we have

$$\dot{\hat{x}}_n - y^{(n)} = -\vartheta \psi^{(n+1)}.$$

From system (7.1), $y^{(n)} = f(\underline{x}, u)$. Substituting it to the above equation gives

$$\dot{\hat{x}}_n = f(\underline{x}, u) - \vartheta \psi^{(n+1)}$$

Let $\Delta f = f(\underline{x}, u) - f(\hat{\underline{x}}, u)$. The above equation becomes

$$\dot{\hat{x}}_n = f(\hat{\underline{x}}, u) + \Delta f - \vartheta \psi^{(n+1)}$$

Thus, the dynamics of the observed states can be represented as

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2 \\ \dot{\hat{x}}_2 &= \hat{x}_3 \\ &\dots \\ &\dots \\ \dot{\hat{x}}_n &= f(\hat{\underline{x}}, u) + \Delta f - \vartheta \psi^{(n+1)} \\ \hat{y} &= \hat{x}_1 \end{aligned} \tag{7.12}$$

The bound of $\Delta f - \vartheta \psi^{(n+1)}$ can be derived as follows. Since $f(\underline{x}, u)$ is Lipschiz, there exists a constant L such that

$$\|f(\underline{x}, u) - f(\hat{\underline{x}}, u)\| \leq L \|\underline{x} - \hat{\underline{x}}\|$$

Moreover, from lemma 7.1

$$\|\underline{x} - \hat{\underline{x}}\| = \vartheta \|\underline{\psi}\| \leq \vartheta \|\underline{h}\| \tag{7.13}$$

where $\underline{\psi} = (\psi^{(1)} \quad \psi^{(2)} \quad \psi^{(3)} \quad \dots \quad \psi^{(n)})^T$, and $\underline{h} = (h_1 \quad h_2 \quad h_3 \quad \dots \quad h_n)^T$. Thus,

$$|f(\underline{x}, u) - f(\hat{\underline{x}}, u)| \leq L \|\underline{x} - \hat{\underline{x}}\| \leq \vartheta L \|\underline{h}\|$$

Therefore, the bound of $\Delta f - \vartheta \psi^{(n+1)}$ is

$$|\Delta f - \vartheta \psi^{(n+1)}| \leq \vartheta L \|\underline{\psi}\| - \vartheta \psi^{(n+1)} \leq \vartheta (L \|\underline{h}\| + h_{n+1}) \leq H \quad (7.14)$$

in which we have used $|\psi^{(n+1)}| \leq h_{n+1}$ and $H = L \|\underline{h}\| + h_{n+1}$ is a bounded positive constant independent from ϑ .

• Step 2:

From theorem 6.1, a controller (7.10) with the self-structuring algorithm described in section 5.1.1 and the adaptive law (7.11) applied to system

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2 \\ \dot{\hat{x}}_2 &= \hat{x}_3 \\ &\dots \\ \dot{\hat{x}}_n &= f(\hat{\underline{x}}, u) \\ \hat{y} &= \hat{x}_1 \end{aligned}$$

will guarantee that

$$\|\hat{\underline{e}}(t)\| \leq \sqrt{\frac{2 \max \left\{ V(0), \frac{1}{2\alpha} \left(\frac{c}{\gamma} + \|P \underline{b}_c\|^2 \|\varepsilon^*\|^2 \right) \right\}}{\lambda_{\min}(P)}}, \quad \forall t > 0,$$

$$\text{and } \hat{\underline{e}}(t) \text{ converges to compact set } \Omega_{\hat{\underline{e}}} = \left\{ \hat{\underline{e}}(t) \mid \|\hat{\underline{e}}(t)\| \leq \sqrt{\frac{\|P \underline{b}_c\|^2 \|\varepsilon^*\|^2}{\lambda_{\min}(Q) - 1}} \right\}.$$

Now, for system (7.12), we can consider $\Delta f - \vartheta \psi^{(n+1)}$ as part of the approximation error. Thus, we have $\varepsilon_{new} = \varepsilon + \Delta f - \vartheta \psi^{(n+1)}$ and $\varepsilon_{new}^* = \varepsilon^* + \vartheta H$, in which H is a bounded constant defined in (7.14). It is straight forward that a controller (7.10) with the self-structuring algorithm proposed in chapter 5 and the adaptive law (7.11) applied to system (7.12) guarantees that

$$\|\hat{\underline{e}}(t)\| \leq \sqrt{\frac{2 \max \left\{ V(0), \frac{1}{2\alpha} \left(\frac{c}{\gamma} + \|P \underline{b}_c\|^2 \|\varepsilon^* + \vartheta H\|^2 \right) \right\}}{\lambda_{\min}(P)}}, \quad \forall t > 0, \quad (7.15)$$

and $\hat{e}(t)$ converges to compact set

$$\Omega_{\hat{e}} = \left\{ \hat{e}(t) \left| \|\hat{e}(t)\| \leq \sqrt{\frac{\|P\underline{b}_c\|^2 \|\varepsilon^* + \varepsilon H\|^2}{\lambda_{\min}(Q) - 1}} \right. \right\}. \quad (7.16)$$

• Step 3:

The actual tracking error is

$$\underline{e} = \underline{r} - \underline{x} = \hat{\underline{e}} - (\underline{x} - \hat{\underline{x}}) = \hat{\underline{e}} - \varepsilon \underline{\psi}$$

Thus, from lemma 7.1,

$$\|\underline{e}\| \leq \|\hat{\underline{e}}\| + \varepsilon \|\underline{\psi}\| \leq \|\hat{\underline{e}}\| + \varepsilon \|\underline{h}\|. \quad (7.17)$$

From (7.15), (7.16), and (7.17),

$$\|\underline{e}(t)\| \leq \sqrt{\frac{2 \max \left\{ V(0), \frac{1}{2\alpha} \left(\frac{c}{\gamma} + \|P\underline{b}_c\|^2 \|\varepsilon^* + \varepsilon H\|^2 \right) \right\}}{\lambda_{\min}(P)}} + \varepsilon \|\underline{h}\|, \quad \forall t > 0, \quad (7.18)$$

and $\underline{e}(t)$ converges to compact set

$$\Omega_{\underline{e}} = \left\{ \underline{e}(t) \left| \|\underline{e}(t)\| \leq \sqrt{\frac{\|P\underline{b}_c\|^2 \|\varepsilon^* + \varepsilon H\|^2}{\lambda_{\min}(Q) - 1}} + \varepsilon \|\underline{h}\| \right. \right\}. \quad (7.19)$$

This concludes the proof. \diamond

Remark 7.1 It should be noted that the choices of the controller's parameters are independent from the choice of the observer's parameters. Thus, it makes the design of the output-feedback SSDAFC scheme two separate steps: design an observer (7.7) for system (7.1), and design a SSDAFC for the observed system (7.10). This preserves the main advantage of using a high-gain observer. In [2], high-gain observers are also employed, but the design of the controller depends on the design of the observer. Thus, the approach is more complicated and parameter tuning is more difficult.

Remark 7.2 As the choices of the controller's parameters are independent from the choice of the observer's parameters, remarks in chapters 4, 5, 6 are still valid

Remark 7.3 Theorem 7.1 shows that choosing a smaller ε will result in smaller tracking error. However, too small ε will result in peaking phenomenon and chattering in transient behaviour. Saturation methods introduced in [59, 60] have been

suggested to overcome this problem. Here, the peaking phenomenon is completely avoided by the use of the adaptive law (7.11).

Remark 7.4 Recently a new non-observer approach has been proposed in [96], in which only the output error is used to generate control input and update laws for unknown fuzzy parameters, and no state observer or low-pass filter is required. In the future, it would be interesting to investigate the possibility to incorporate our proposed self-structuring algorithm with this approach.

7.5. Example

7.5.1. Continuously stirred tank reactor (CSTR) system without zero dynamics

We consider the CSTR system given in [2]. This system consists of a constant volume reactor cooled by a single coolant stream. An irreversible, exothermic reaction, $A \rightarrow B$, occurs in the tank. The objective is to control the concentration C_a by manipulating the coolant flow rate q_c . The process is described by the following differential equations

$$\begin{aligned}\dot{C}_a &= \frac{q}{V}(C_{a0} - C_a) - a_0 C_a e^{-\frac{E}{R.T_a}} \\ \dot{T}_a &= \frac{q}{V}(T_f - T_a) + a_1 C_a e^{-\frac{E}{R.T_a}} + a_3 q_c \left[1 - e^{-\frac{a_2}{q_c}} \right] (T_{cf} - T_a)\end{aligned}$$

where C_a and T_a are the concentration and temperature of the tank, respectively; the coolant flow rate q_c is the control input; and the parameters of the system are given in table 7.1.

Parameter	Description	Nominal value
q	Process flow rate	100l/min
C_{a0}	Concentration of component A	1mol/l
T_f	Feed temperature	350K
T_{cf}	Inlet coolant temperature	350K
V	Volume of tank	100l
h_a	Heat transfer coefficient	$7 \times 10^5 J / \text{min} \cdot K$
a_0	Pre-exponential factor	$7.2 \times 10^{10} \text{ min}^{-1}$
E/R	Activation energy	$1 \times 10^4 K$
$(-\Delta H)$	Heat of reaction	$2 \times 10^4 \text{ cal} / \text{mol}$
ρ_1, ρ_2	Liquid densities	$1 \times 10^3 \text{ g} / \text{l}$
C_p, C_{pc}	Heat capacities	$1 \text{ cal} / \text{g} \cdot K$
$a_1 = 1.44 \times 10^{13}$	$a_2 = 6.987 \times 10^2$	$a_3 = 0.01$

Table 7.1: parameters of the CSTR system

Define the state variables, input, and output as

$$\underline{x} = [x_1, x_2]^T = [C_a, T_a]^T, \quad u = q_c, \quad y = C_a$$

The dynamics system can be written in the form of system (7.2)

$$\dot{\underline{x}} = \underline{f}(\underline{x}, u) = \begin{bmatrix} 1 - x_1 - a_0 x_1 e^{-\frac{10^4}{x_2}} \\ 350 - x_2 + a_1 x_1 e^{-\frac{10^4}{x_2}} + a_3 u \left(1 - e^{-\frac{a_2}{u}} \right) (350 - x_2) \end{bmatrix}$$

$$y = h(\underline{x}) = x_1$$

From the parameters in table 7.1 and the irreversible exothermic property of the chemical process, we obtain the operating region of the states and control input as follows

$$0 < x_1 < 1, \quad h_1 \geq x_2 > 350, \quad 0 \leq u \leq h_2$$

where constant h_1 is the highest temperature of the reactor and constant h_2 is the maximum value of the coolant flow rate.

We have

$$\dot{y} = L_f h = 1 - x_1 - a_0 x_1 e^{-\frac{10^4}{x_2}}, \quad \frac{\partial [L_f h]}{\partial u} = 0,$$

$$\ddot{y} = L_f^2 h = -\dot{x}_1 - a_0 \left(\dot{x}_1 + \frac{10^4 x_1 \dot{x}_2}{x_2^2} \right) e^{-\frac{10^4}{x_2}},$$

and

$$\frac{\partial[L_f^2 h]}{\partial u} = 10^4 a_0 a_3 x_1 e^{-\frac{10^4}{x_2}} \frac{(x_2 - 350)}{x_2^2} \left(1 - e^{-\frac{a_2}{u}} - \frac{a_2}{u} e^{-\frac{a_2}{u}} \right) > 0$$

by using the fact that $1 - e^{-w} - we^{-w} > 0, \forall w > 0$. Therefore, the plant is of relative degree 2 and assumption 6.1 is satisfied. Using the transformation given in 7.2.1, the system can be represented as

$$\begin{cases} \dot{\xi}_1 = \xi_2 \\ \dot{\xi}_2 = b(\underline{\xi}, u) \\ y = \xi_1 \end{cases}$$

where $\xi_1 = x_1$, $\xi_2 = \dot{x}_1$, $b(\underline{\xi}, u) = L_f^2 h(\underline{x})$. The above analysis is just to check the validity of the assumptions made. Now, we can use the output feedback SSAFC in theorem 7.1 to control the system without the knowledge of the mathematical model of the CSTR.

For comparison purpose, we choose the same control objective as in [2]. The control objective is to make the concentration C_a track the set-point $r(t)$ of $\pm 0.02 \text{ mol/l}$ about the nominal product concentration of 0.1 mol/l . The initial conditions are chosen as the nominal operating conditions $\underline{x}_0 = [0.1, 438.5]^T$.

The control input is chosen as (7.10):

$$u = \hat{u}(\hat{X} | \theta_{ac}) = \sum_{j=1}^M \theta_j \zeta_j(\hat{X}) \text{ in which } \hat{X} = [\hat{\xi}_1, \hat{\xi}_2, \hat{v}]^T.$$

The operating variable ranges are chosen as follows:

$$\xi_1 \in [0, 0.2]; \xi_2 \in [-0.5, 0.5]; v \in [-1, 1].$$

The controller parameters are chosen as follows:

$$\underline{k} = [1 \quad 1]^T; Q = \begin{bmatrix} 20 & 0 \\ 0 & 10 \end{bmatrix}; P = \begin{bmatrix} 25 & 10 \\ 10 & 15 \end{bmatrix}; \gamma = 1000;$$

$$u_L = 0; u_U = 500.$$

The structure-learning parameters are as follows:

$$\varepsilon_0 = 0.5, \text{ error_threshold} = 1.5, \text{ max_mf_distance} = [0.2 \quad 2 \quad 2]$$

$$\text{min_mf_distance} = [0.2 \quad 2 \quad 2]/4, B_{rules} = 50.$$

The initial fuzzy system has 8 rules with 2 membership functions in each input dimension as shown in fig 7.1a-7.1c. Using the expert knowledge that at nominal

condition, the control input is “near” $100\text{mol}/l$, we initialize the consequents to $100\text{mol}/l$.

Since the time derivative of output y is not available, it is estimated as proposed in theorem 7.1 by a 2nd-order high-gain observer:

$$\begin{cases} \hat{\xi}_1 = \pi_1 \\ \hat{\xi}_2 = \frac{\pi_2}{\vartheta} \end{cases} \quad \text{and} \quad \begin{cases} \vartheta \dot{\pi}_1 = \pi_2 \\ \vartheta \dot{\pi}_2 = -\bar{\lambda}_1 \pi_2 - \pi_1 + y(t) \end{cases} \quad \text{with } \bar{\lambda}_1 = 1, \vartheta = 0.1, \text{ and the initial}$$

condition $[\hat{y}_1(0), \hat{y}_2(0)]^T = [0.1, 0]^T$.

The simulation results are shown in Fig 7.2a-7.2e. It can be seen in Fig 7.2a that the concentration tracks the desired reference well. The control signal is in the desired range $[0, 500]$ (Fig 7.2b). Fig 7.2c and Fig 7.2d show the state estimation errors. Fig 7.2e shows the number of rules and structure learning flag. It can be seen that no structure learning is required in this case.

For comparison, the results of the adaptive multi-layer NN controller and the fixed-gain proportional plus integral (PI) controller given in [2] are shown in Fig 7.3. It can be observed that our controller is also better than the PI controller. The multi-layer NN controller is slightly better than ours. This is due to the addition of a PI control term and a robust control term in the multi-layer NN controller. Our controller is relatively simpler as it has only one control term, the output of the fuzzy system. By incorporating expert knowledge to initialize the consequents to $100\text{mol}/l$, the set-point tracking of our controller is still guaranteed during the initial period without the use of PI and robust control terms. This demonstrates an advantage of adaptive fuzzy control over adaptive NN control, the ability to incorporate expert knowledge to initialize controllers.

7.5.2. Continuously stirred tank reactor (CSTR) system with zero dynamics

We consider the CSTR system presented in [65]. A class of multi-component isothermal reaction $A \leftrightarrow B \rightarrow C$ is taking place in the reactor. The output of the process is the concentration of A , and the manipulated variable is the molar feed flow rate of B , N_{BF} . A mass balance gives the modelling equations:

$$\begin{aligned}
V \frac{dC_A}{dt'} &= F(C_{A_f} - C_A) - Vk_1 C_A + Vk_2 C_B^2 \\
V \frac{dC_B}{dt'} &= -FC_B - Vk_1 C_B^2 - Vk_3 C_B^2 + N_{BF} \\
V \frac{dC_C}{dt'} &= -FC_C + Vk_3 C_B^2 \\
y &= C_A
\end{aligned}$$

With the dimensionless variables given in table 7.2, we can obtain the dimensionless state-space model description:

$$\begin{aligned}
\dot{x}_1 &= 1 - x_1 - c_1 x_1 + c_2 x_2^2 \\
\dot{x}_2 &= -x_2 + c_1 x_1 - c_2 x_2^2 - c_3 x_2^2 + u \\
\dot{x}_3 &= -x_3 + c_3 x_2^2 \\
y &= x_1
\end{aligned}$$

It is easy to check that the relative degree of this system is 2. Using the transformation given in 7.2.1, the system can be transformed into

$$\begin{aligned}
\dot{\xi}_1 &= \xi_2 \\
\dot{\xi}_2 &= f_0(\xi_1, \xi_2) + g_0(\xi_1, \xi_2)u \\
\dot{\eta} &= -\eta + c_3 f_t
\end{aligned}$$

where

$$\begin{aligned}
f_t &= [(1 + c_1)\xi_1 + \xi_2 - 1]/c_2, \\
f_0(\xi_1, \xi_2) &= 2c_1 c_2 \sqrt{f_t} \xi_1 - (c_1 + 1)\xi_2 - 2c_2 [1 + (c_2 + c_3)\sqrt{f_t}] f_t, \\
g_0(\xi_1, \xi_2) &= 2c_2 \sqrt{f_t}. \text{ Also, the Damkholer numbers are assumed as follows: } c_1 = 20, \\
c_2 &= 0.1, \text{ and } c_3 = 10. \text{ Now, we can use the output feedback SSDAFC in theorem 7.1} \\
&\text{to control this system.}
\end{aligned}$$

For comparison purposes, we choose the same control objective as in [65]. The control objective is to make the concentration C_A track the set-point $r(t)$ of ± 0.02 about the nominal product concentration of 0.1. The initial conditions are chosen as the nominal operating conditions $\underline{x}_0 = [0.1, 3.3, 110]^T$.

The control input is chosen as (7.10):

$$u = \hat{u}(\hat{X}|\theta_{ac}) = \sum_{j=1}^M \theta_j \zeta_j(\hat{X}) \text{ in which } \hat{X} = [\hat{\xi}_1, \hat{\xi}_2, \hat{v}]^T.$$

The operating variable ranges are chosen as follows:

$$\xi_1 \in [0.05, 0.15]; \xi_2 \in [-0.05, 0.05]; v \in [-0.8, 0.8].$$

The controller parameters are chosen as follows:

$$\underline{k} = [1 \quad 10]^T; Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}; P = \begin{bmatrix} 51 & 15 \\ 5 & 1 \end{bmatrix}; \gamma = 1000;$$

$$u_L = 0; u_U = 500.$$

The structure-learning parameters are as follows:

$$\varepsilon_0 = 0.5, \text{ error_threshold} = 1.5, \mathbf{max_mf_distance} = [0.1 \quad 0.1 \quad 1.6],$$

$$\mathbf{min_mf_distance} = [0.1 \quad 0.1 \quad 1.6]/4, B_{rules} = 50.$$

The initial fuzzy system has 8 rules with 2 membership functions in each input dimension as shown in fig 7.4a-7.4c. Using the expert knowledge that at nominal condition, the control input is “near” 100, we initialize the consequents to 100.

Since the time derivative of output y is not available, it is estimated as proposed in theorem 7.1 by a 2nd-order high-gain observer:

$$\begin{cases} \hat{\xi}_1 = \pi_1 \\ \hat{\xi}_2 = \frac{\pi_2}{\vartheta} \end{cases} \text{ and } \begin{cases} \vartheta \dot{\pi}_1 = \pi_2 \\ \vartheta \dot{\pi}_2 = -\bar{\lambda}_1 \pi_2 - \pi_1 + y(t) \end{cases} \text{ with } \bar{\lambda}_1 = 1, \vartheta = 0.1, \text{ and the initial}$$

condition $[\hat{y}_1(0), \hat{y}_2(0)]^T = [0.1, 0]^T$.

The simulation results are shown in Fig 7.5a-7.5g. It can be seen in Fig 7.5a that the concentration tracks the desired reference well. The control signal is in the desired range $[0, 500]$ (Fig 7.5b). Fig 7.5c and Fig 7.5d show the state estimation errors. Fig 7.5e shows the number of rules and structure learning flag. It can be seen that no structure learning is required in this case. Fig 7.5f shows the internal dynamics. It can be seen that the internal dynamics is stable.

The tracking performance obtained in [65] is given in Fig 7.6. It can be seen that both Ge’s controller and our controller are successful. Ge’s controller responds faster. This maybe due to the addition of an a priori control term based on a nominal model and a bounding control term. Our controller is relatively simpler as it has only one control term, the output of the fuzzy system. Also, Ge’s controller requires 500 neurons, whereas our controller requires only 8 rules.

7.5.3. Third-order system in triangular form (7.3)

A third-order system is given in [2] as:

$$\begin{aligned}
\dot{x}_1 &= x_1 + x_2 + \frac{x_2^3}{5} \\
\dot{x}_2 &= x_1 x_2 + (2 + \sin(x_1 x_2))x_3 \quad \text{with } [x_1(0), x_2(0), x_3(0)]^T = [1.4, 0.3, 0.1]^T. \\
\dot{x}_3 &= x_2 x_3 + (1 + x_1^2 + x_2^2)u \\
y &= x_1
\end{aligned}$$

The control objective is to make the output of the system track the desired trajectory y_d generated from the Van der Pol oscillator:

$$\begin{aligned}
\dot{x}_{d1} &= x_{d2} \\
\dot{x}_{d2} &= -x_{d1} + \beta(1 - x_{d1}^2)x_{d2} \quad \text{with } [x_{d1}(0), x_{d2}(0)]^T = [1.5, 0.2]^T \text{ and } \beta = 0.2. \\
y_d &= x_{d1}
\end{aligned}$$

It can be seen that the system satisfies assumption 7.2, thus it can be transformed to the form (7.4) :

$$\begin{aligned}
\dot{\xi}_1 &= \xi_2 \\
\dot{\xi}_2 &= \xi_3 \\
\dot{\xi}_3 &= b(\xi, u) \\
y &= \xi_1
\end{aligned}$$

Now, we use the output feedback SSDAFC in theorem 7.1 to control this system. As this is a 3rd-order system, the desired output and its derivatives ($y_d, \dot{y}_d, \ddot{y}_d, \ddot{\ddot{y}}_d$) are needed. ($y_d, \dot{y}_d, \ddot{y}_d, \ddot{\ddot{y}}_d$) can be estimated from $(x_{d1}, x_{d2})^T$ as follows:

$$\begin{aligned}
y_d &= x_{d1} \\
\dot{y}_d &= x_{d1} = x_{d2} \\
\ddot{y}_d &= \dot{x}_{d2} = -x_{d1} + \beta(1 - x_{d1}^2)x_{d2} \\
\ddot{\ddot{y}}_d &= -x_{d2} + \beta(1 - 2x_{d1}x_{d2})x_{d2} + \beta(1 - x_{d1}^2)(-x_{d1} + \beta(1 - x_{d1}^2)x_{d2})
\end{aligned}$$

Since the time derivative of output y is not available, it is estimated as proposed in theorem 7.1 by a 3rd-order high-gain observer:

$$\begin{cases} \hat{\xi}_1 = \pi_1 \\ \hat{\xi}_2 = \frac{\pi_2}{\vartheta} \\ \hat{\xi}_3 = \frac{\pi_2}{\vartheta} \end{cases} \quad \text{and} \quad \begin{cases} \vartheta \dot{\pi}_1 = \pi_2 \\ \vartheta \dot{\pi}_2 = \pi_3 \\ \vartheta \dot{\pi}_3 = -\bar{\lambda}_1 \pi_3 - \bar{\lambda}_2 \pi_2 - \pi_1 + y(t) \end{cases} \quad \text{with } \bar{\lambda}_1 = \bar{\lambda}_2 = 3, \vartheta = 0.2, \text{ and}$$

the initial condition $[\hat{y}_1, \hat{y}_2, \hat{y}_3] = [1.4, 1.7, 2.4]^T$.

The control input is chosen as (7.10):

$$u = \hat{u}(\hat{X}|\theta_{ac}) = \sum_{j=1}^M \theta_j \zeta_j(\hat{X}) \text{ in which } \hat{X} = [\hat{\xi}_1, \hat{\xi}_2, \hat{v}]^T.$$

The operating variable ranges are chosen as follows:

$$\xi_1 \in [-5, 5]; \xi_2 \in [-3, 3]; \xi_3 \in [-3, 3] \quad v \in [-3, 3].$$

The controller parameters are chosen as follows:

$$\underline{k} = [5 \quad 50 \quad 10]^T; \underline{Q} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}; \underline{P} = \begin{bmatrix} 510.86 & 103.59 & 10 \\ 103.59 & 47.58 & 2.17 \\ 10 & 2.17 & 0.72 \end{bmatrix}; \gamma = 50;$$

$$u_L = -50; u_U = 50.$$

The structure-learning parameters are as follows:

$$\varepsilon_0 = 0.5, \text{ error_threshold} = 2.5, \text{max_mf_distance} = [10 \quad 6 \quad 6 \quad 6]$$

$$\text{min_mf_distance} = [10 \quad 6 \quad 6 \quad 6]/5, B_{rules} = 200.$$

The initial fuzzy system has 16 rules with 2 membership functions in each input dimension as shown in fig 7.7a-7.7b. All the consequents are initialized to 0.

The simulation results are given in Fig 7.8. It can be seen in Fig 7.8a that the output tracks the reference signal well. The control signal is in the range $[-50, 50]$. The actual states and their estimations are shown in Fig 7.8c-7.8e. the number of rules and structuring flag is shown in Fig 7.8f. It can be seen that the final fuzzy controller has 54 rules. The final membership functions of ξ_1, ξ_2, ξ_3, v are shown in Fig 7.9.

Compared to the adaptive NN controller proposed in [2], our controller has similar performance. However, our controller requires only 1 fuzzy system with 54 rules, whereas the adaptive NN controller requires 3 neural networks (64 nodes, 256 nodes, 1024 nodes). Moreover, the adaptive NN controller also requires calculations of some partial derivatives, which increase the complexity of the controller.

7.6. Conclusion

In this chapter, we extend the control scheme in chapter 6 to two broader classes of nonlinear systems:

$$\begin{aligned} \underline{x} &= \underline{f}(\underline{x}, u) \\ y &= \underline{h}(\underline{x}) \end{aligned},$$

$\dot{x}_1 = f_1(x_1, \dots, x_{i+1}) \quad i = 1, \dots, n-1$
 and $\dot{x}_n = f_n(x_1, \dots, x_n, u)$. We show that these classes can be
 $y = x_1$

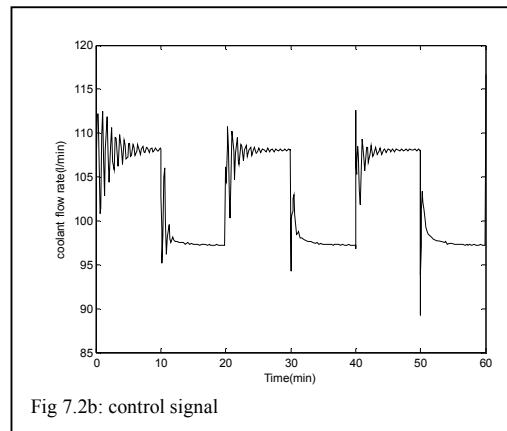
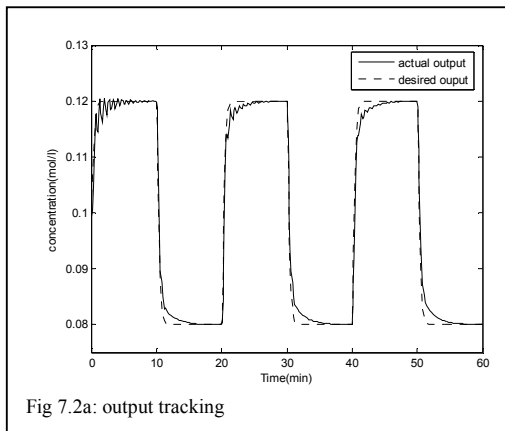
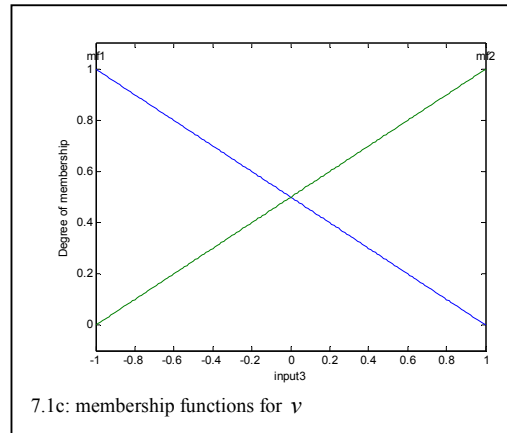
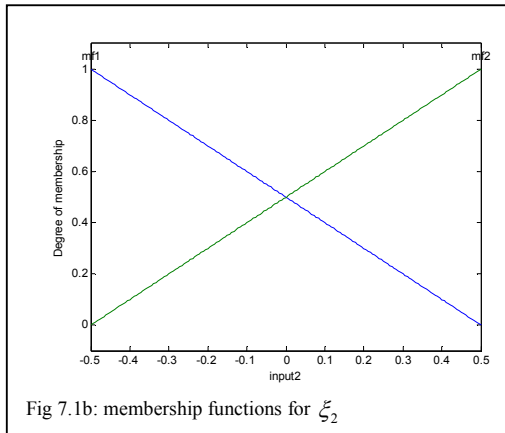
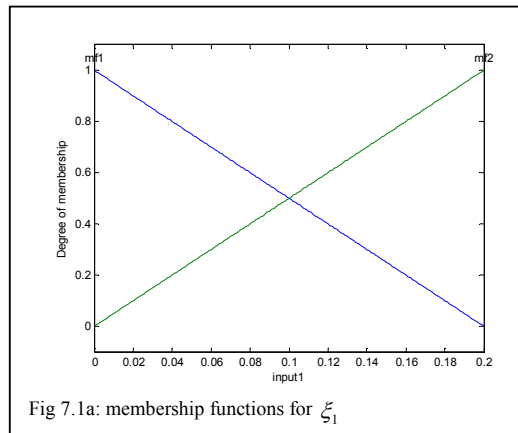
transformed to the nominal form

$$\begin{cases} \dot{\xi}_i = \xi_{i+1}, i = 1, \dots, \rho - 1 \\ \dot{\xi}_\rho = b(\underline{\xi}, \underline{\eta}, u) \\ y = \xi_1 \end{cases}$$

Thus, the control scheme proposed in chapter 6 can apply to these classes.

In case the derivatives of y are not available, we propose output feedback SSDAFC using high-gain observers. Application to 3 nonlinear systems demonstrates the effectiveness of the output feedback SSDAFC scheme.

In the next chapter, we will present the software implementation of control schemes proposed so far.



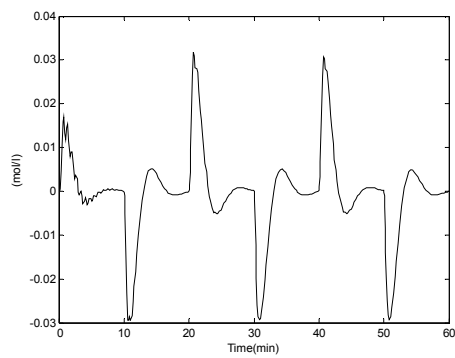


Fig 7.2c: state estimation error $x_1 - \hat{z}_1$

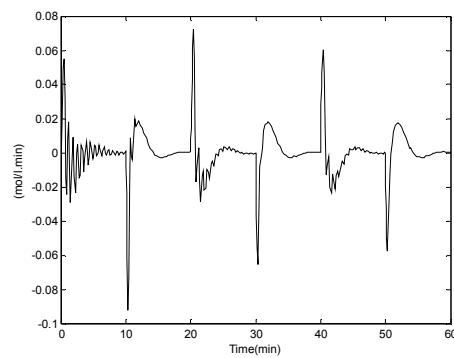


Fig 7.2d: state estimation error $x_1 - \hat{z}_{12}$

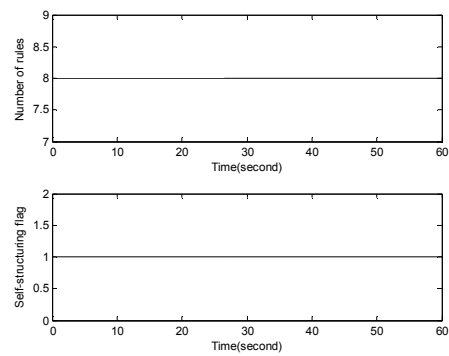
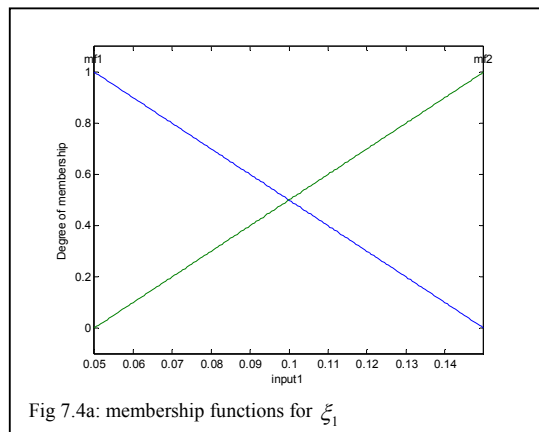
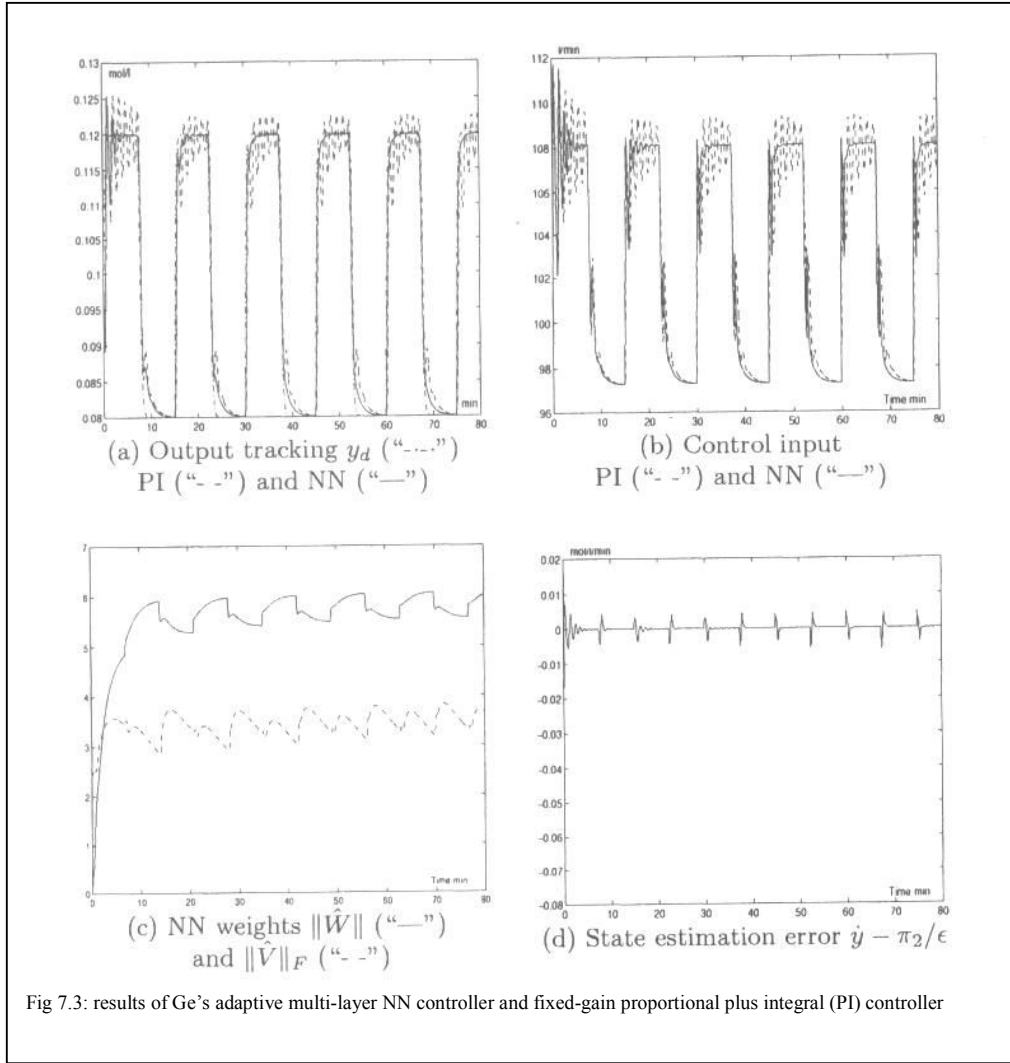
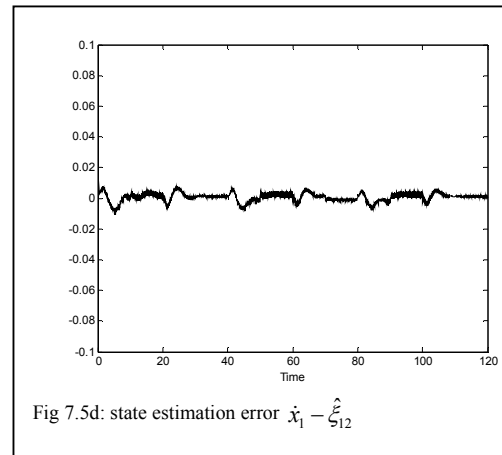
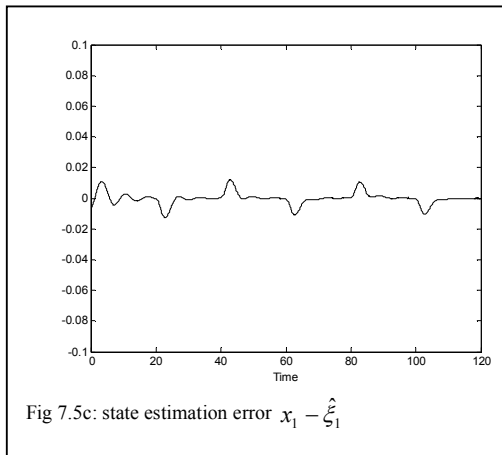
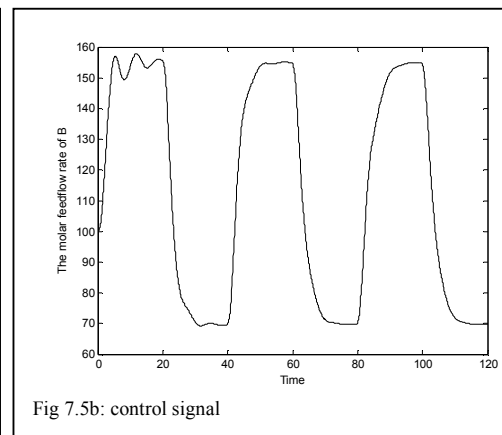
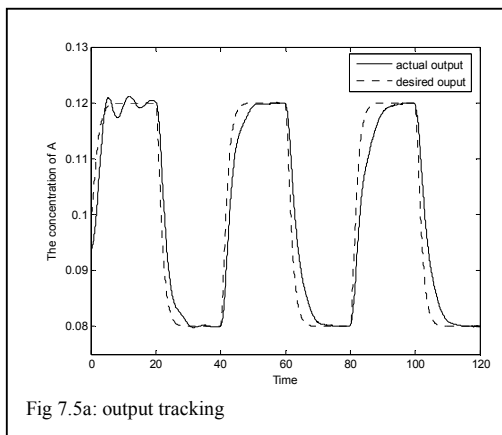
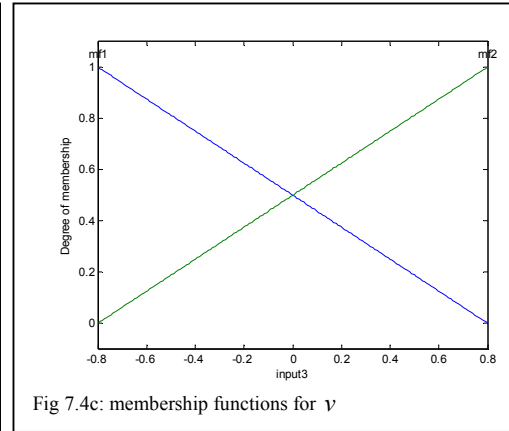
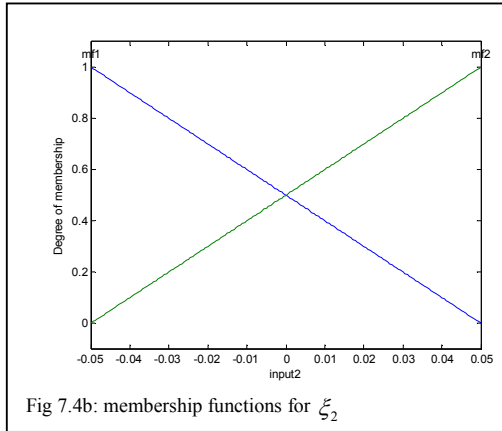
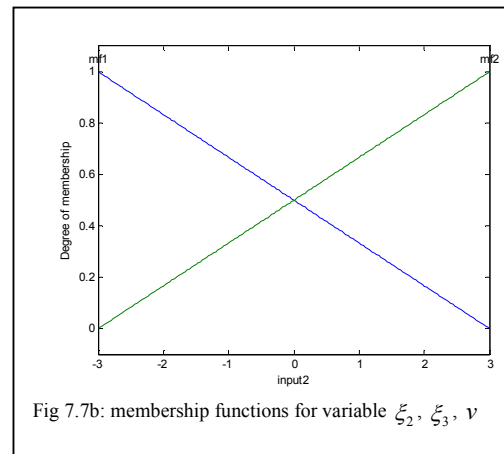
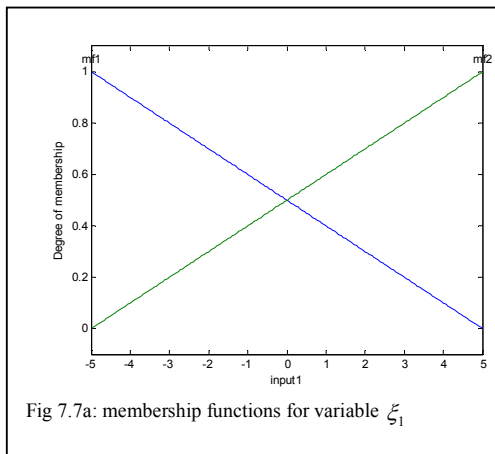
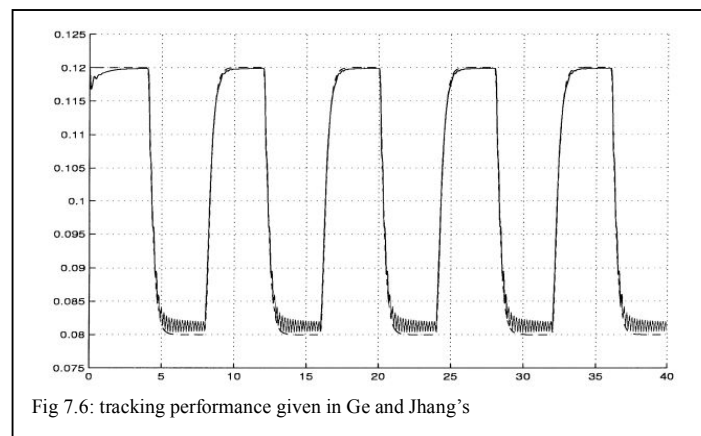
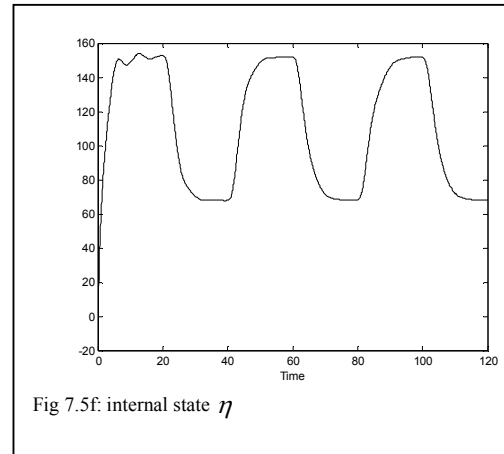
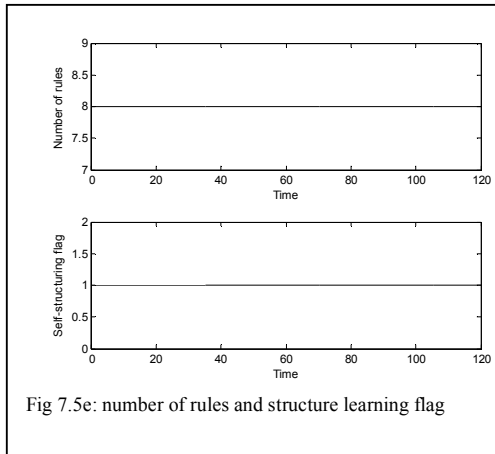
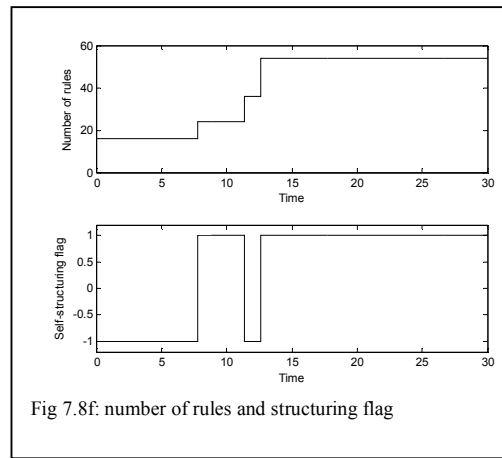
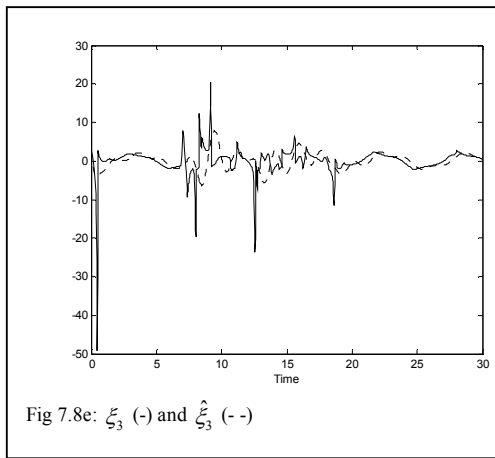
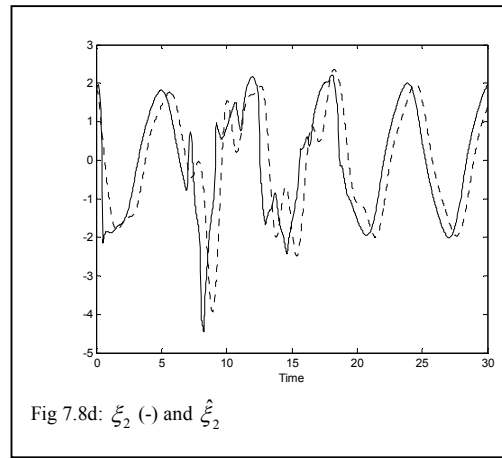
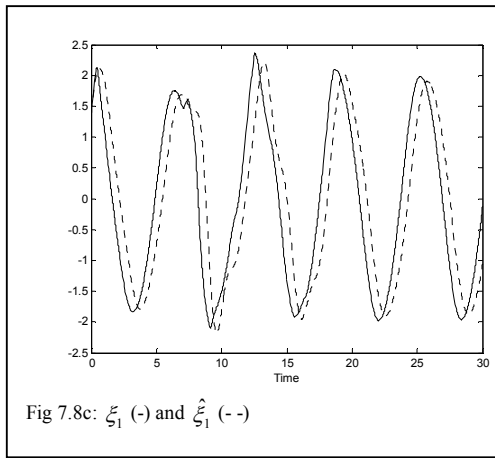
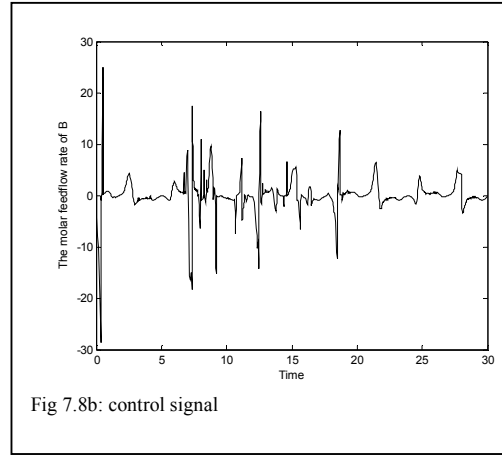
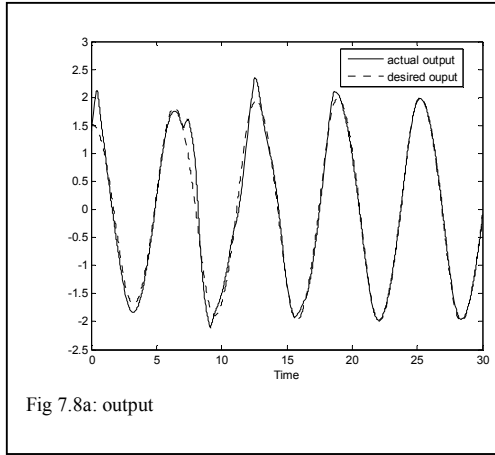


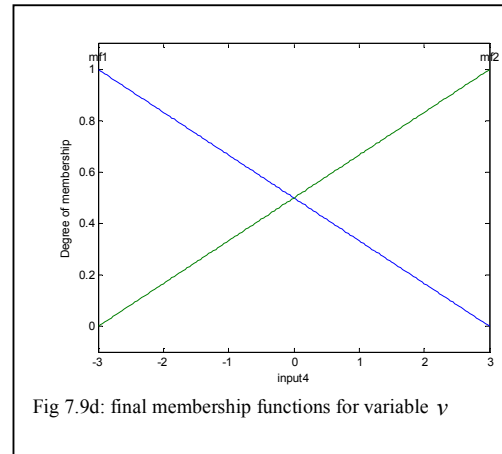
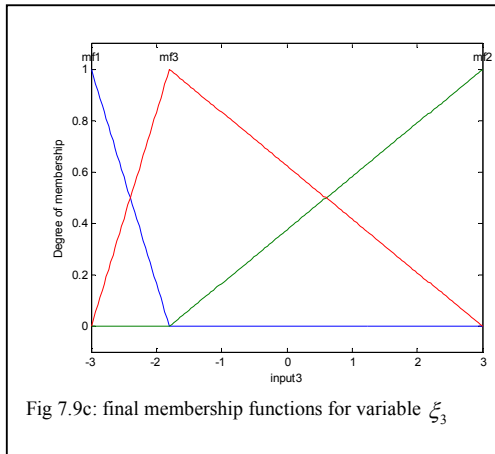
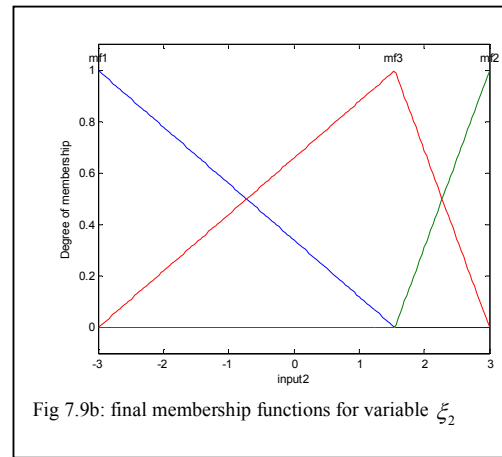
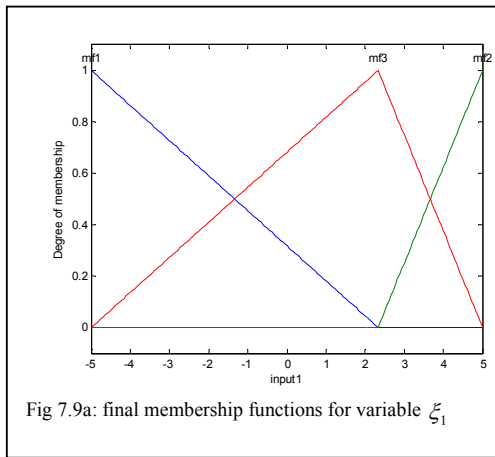
Fig 7.2e: number of rules and structure learning flag











8. Chapter 8

MATLAB IMPLEMENTATION

8.1. Introduction

Matlab and Simulink are integrated software packages that maybe used for modeling, simulating, and analyzing dynamic systems. Simulink provides a graphical user interface (GUI) for building models as block diagrams, using click-and-drag mouse operations. It is a powerful tool for Simulation and Model-Based Design. Simulink applications range from control design, signal processing and communications, image processing, etc. [97].

In control design area, designers use Simulink and add-on products to design and create software that is used in aerospace, defense, automotive, industrial equipment, process control, and many other applications. Thus, Simulink is a very suitable tool to implement our control algorithms.

In this chapter, we present the software implementation of our proposed control algorithms. Programming issues are discussed in section 8.2. Section 8.3 presents our Adaptive Fuzzy Control simulink library, which includes a DAFC block, a SSDAFC block, and a high-gain observer block. Then, the simulation process is explained through an example of controlling an inverted pendulum. Real-time control is discussed in section 8.5.

8.2. Programming

The implementation of the developed control algorithms required extensive coding of programs and functions. All of the programs and functions were written using M-language (Matlab script). Custom fuzzy functions also had to be developed as the standard Matlab fuzzy toolbox is not sufficient. All Simulink blocks were built using 2-level M-file S-function template. For clarity, details of the written programs and blocks will not be presented here. We would like to emphasize on the practical use of the developed software.

Next, we will describe the available controller blocks and how to use them for simulation and real time control of dynamic systems.

8.3. Adaptive Fuzzy Control simulink library

Fig 8.1 shows the developed Adaptive Fuzzy Control simulink library, which is ready to be used for control applications. By dragging these blocks to a simulink window, and connect them with a plant (represented by another simulink block), we have a control application ready for simulation. The simulation process will be presented in more detail in section 8.4.

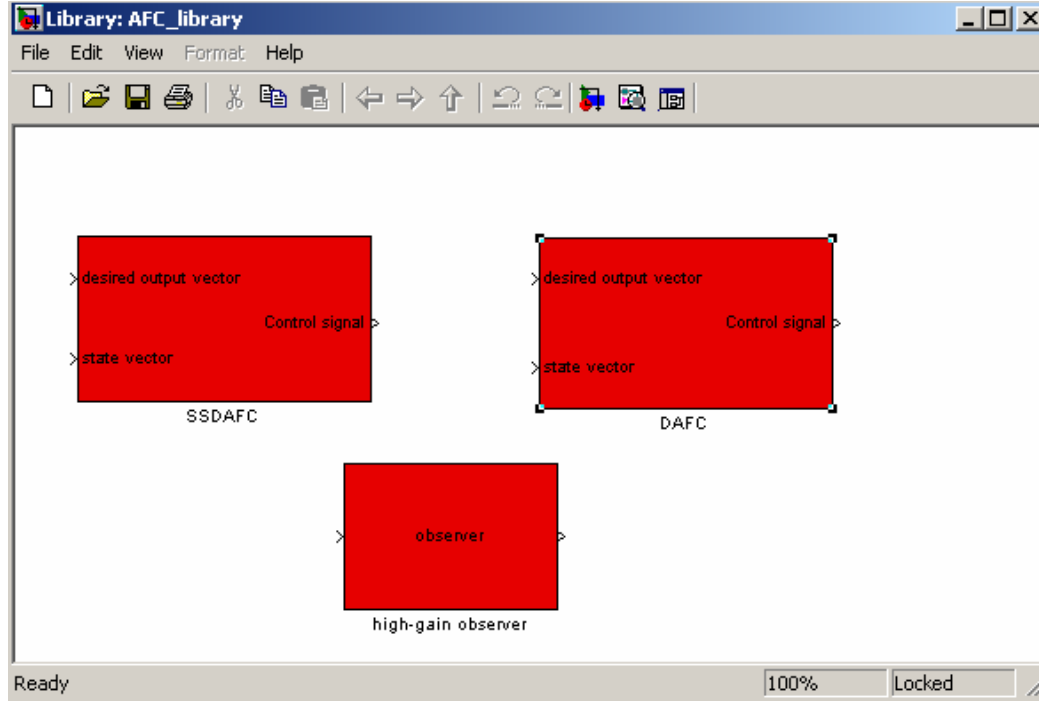


Fig 8.1: Adaptive Fuzzy Control simulink library.

8.3.1. DAFC block

Function:

This block implements the fixed-structured direct adaptive fuzzy controller proposed in theorem 4.1.

Input:

There are two inputs: the state vector \underline{x} and the desired output vector \underline{r} .

$$\underline{x} = (x_1, x_2, \dots, x_n) = (y, \dot{y}, \dots, y^{(n-1)})$$

$$\underline{r} = (r, \dot{r}, \dots, r^{(n-1)})$$

Output:

The output is the control signal generated by the controller.

Parameters:

There are 8 parameters for this controller. These parameters are loaded to the workspace by running the m file “parameters_D AFC.m”.

The code of this file is:

```
%=====
% Parameters of the direct adaptive fuzzy controller %
%=====

% System order
n=2;

%=====load fuzzy controller=====
fuzzy_u=readfis('fuzzy_system_1');

% =====parameters k, A, Q, bc=====
k=[1;1];
A=[0 1;-k(1) -k(2)];
Q=[20 0;0 10];
bc=[0;1];

%=====adaptive law's parameters=====
AFS_params.gamma=25;
AFS_params.theta_U=25;
AFS_params.theta_L=-25;
```

The meaning of the 8 parameters are:

- n: defines the system order.
- fuzzy_u: defines the initial fuzzy controller.

The initial fuzzy system is built using the Matlab fuzzy toolbox and saved to hard-drive under the name “fuzzy_system_1”. When we run the m file “parameters_D AFC.m”, “fuzzy_system_1” will be loaded as the initial fuzzy controller. By changing the name of the file to be loaded (e.g. “fuzzy_system_2”, or “fuzzy_system_3”), we can specify different fuzzy systems as the initial fuzzy controller.

- k: defines vector \underline{k} in equation (4.1).
- A: defines matrix Λ_C in equation (4.6).
- Q: defines matrix Q in equation (4.7).
- bc: defines vector \underline{b}_C in equation (4.6).
- AFS_params.sigma: defines the adaptive gain γ in equation (4.8).

- AFS_params.theta_U: defines u_U in assumption 4.3 and equation (4.8).
- AFS_params.theta_L: defines u_L in assumption 4.3 and equation (4.8).

8.3.2. SSDAFC block

Function:

This block implements the self-structuring direct adaptive fuzzy controller proposed in theorems 5.1 and 6.1.

Input:

There are two inputs: the state vector \underline{x} and the desired output vector \underline{r} .

$$\underline{x} = (x_1, x_2, \dots, x_n) = (y, \dot{y}, \dots, y^{(n-1)})$$

$$\underline{r} = (r, \dot{r}, \dots, r^{(n-1)})$$

Output:

The output is the control signal generated by the controller.

Parameters:

The parameters can be loaded to the workspace by running the file “parameters_SSDAFC.m”.

The code of this file is:

```
%=====
% Parameters of the self-structuring direct adaptive fuzzy controller %
%=====

% System order
n=2;

%=====load fuzzy controller=====
fuzzy_u=readfis('fuzzy_system_1');

% =====parameters k, A, Q, bc=====
k=[1;1];
A=[0 1;-k(1) -k(2)];
Q=[20 0;0 10];
bc=[0;1];

%=====adaptive law's parameters=====
AFS_params.gamma=25;
AFS_params.theta_U=25;
AFS_params.theta_L=-25;
%=====Structure Learning parameters=====
structure_learning_params.max_N_rules = 125;
```



```

structure_learning_params.mf_threshold = 0.5;
structure_learning_params.error_threshold = 5;
structure_learning_params.a0 = [2.5 7.5 10];
structure_learning_params.center_distance_threshold = [1 15/5 20/5];

```

The first 8 parameters have the same meaning as the ones in the DAFC block. The additional 5 parameters are for the self-structuring algorithm. Their meanings are:

- `structure_learning_params.max_N_rules`: defines B_{rule} in assumption 5.1 and assumption 6.3.
- `structure_learning_params.mf_threshold`: defines the completeness of fuzzy rules ε_0 .
- `structure_learning_params.error_threshold`: defines the minimum level of error to trigger structure change *error_threshold*.
- `structure_learning_params.a0`: defines **max_mf_distance**, i.e. the maximum allowed distance between two neighbouring membership functions.
- `structure_learning_params.center_distance_threshold`: defines **min_mf_distance**, i.e. the minimum allowed distance between two neighbouring membership functions.

8.3.3. High-gain observer block

Function:

This block implements the high gain observer given in lemma 7.1.

Input:

The input is the output y of the controlled plant.

Output:

The output is the estimated state vector $\hat{x} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$.

Parameters:

The parameters are loaded to the workspace by running the m file “observer_parameters.m”.

The code of the file “observer_parameters.m” is:\

```

%=====
%   High-gain observer parameters   %
%=====

```

```

lamda_vector = 2; % defines vector lamda
epsilon = 0.5; % defines parameter epsilon

```

Their meanings are:

- lamda_vector: defines vector $\bar{\lambda} = (\bar{\lambda}_1, \bar{\lambda}_2, \dots, \bar{\lambda}_{n-1})^T$ in lemma 7.1.
- epsilon: defines ϵ in lemma 7.1

8.4. AFC simulation

This section presents how an AFC simulation is created and simulated. The simulation process has four steps. This will be presented through an example of controlling an inverted pendulum. Assuming that the state variables are available for measurement, we will employ the fixed-structured DAFC block.

8.4.1. Create a Simulink model for the AFC application

First, we create a new simulink model and add the block representing the inverted pendulum as shown in Fig 8.2.

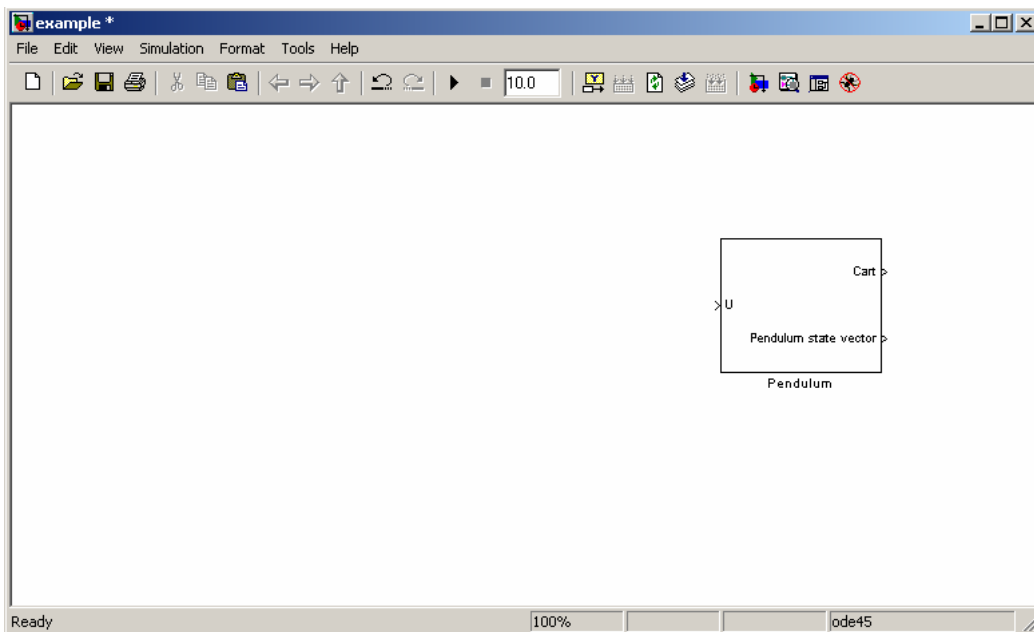


Fig 8.2: The created simulink model with an inverted pendulum block.

Then, drag the DAFC block from the AFC library to the new model and connect with the inverted pendulum as shown in Fig 8.3. The sinusoidal reference signal is generated by the “signal generator” block.

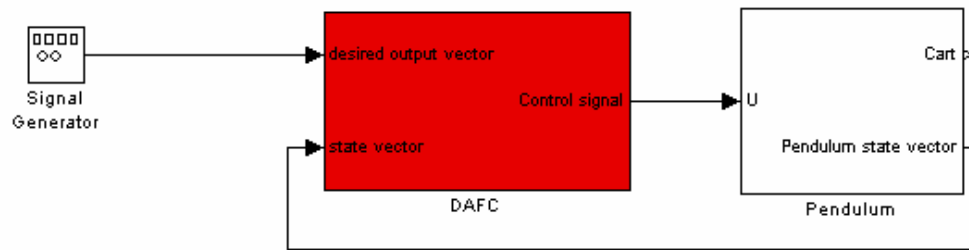


Fig 8.3: Simulink model after the DAFC block and Signal Generator block are added.

To observe the control signal and the output of the inverted pendulum, Scope1 and Scope2 are added as shown in Fig 8.4.

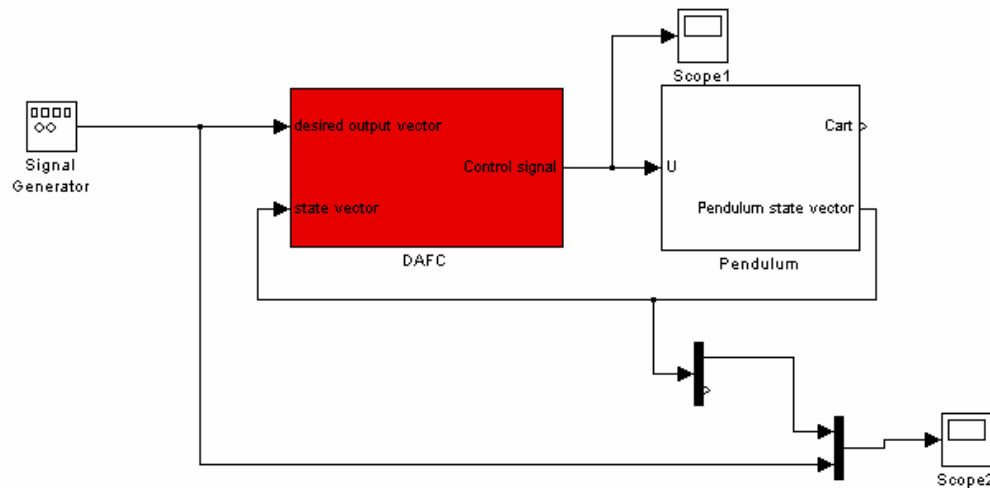


Fig 8.4: Simulink model after added scope1 and scope2.

8.4.2. Design the fuzzy system that is used as the initial controller

The Matlab fuzzy toolbox is used to create the initial fuzzy controller. This fuzzy toolbox allows users to easily create a fuzzy system through its Graphical User Interface (GUI) as demonstrated in Fig 8.5.

As all possible rules are used, it is sometime impractical to add rules one by one, especially for a large number of rules (>50). We have created a script “generate_rules.m” to help add all the possible rules to a fuzzy system. By using the Matlab comment:

```
fuzzy_system1 = generate_fis(fuzzy_system1,b),
```

we add all the possible rules to fuzzy_system1 and initialize them to value b.

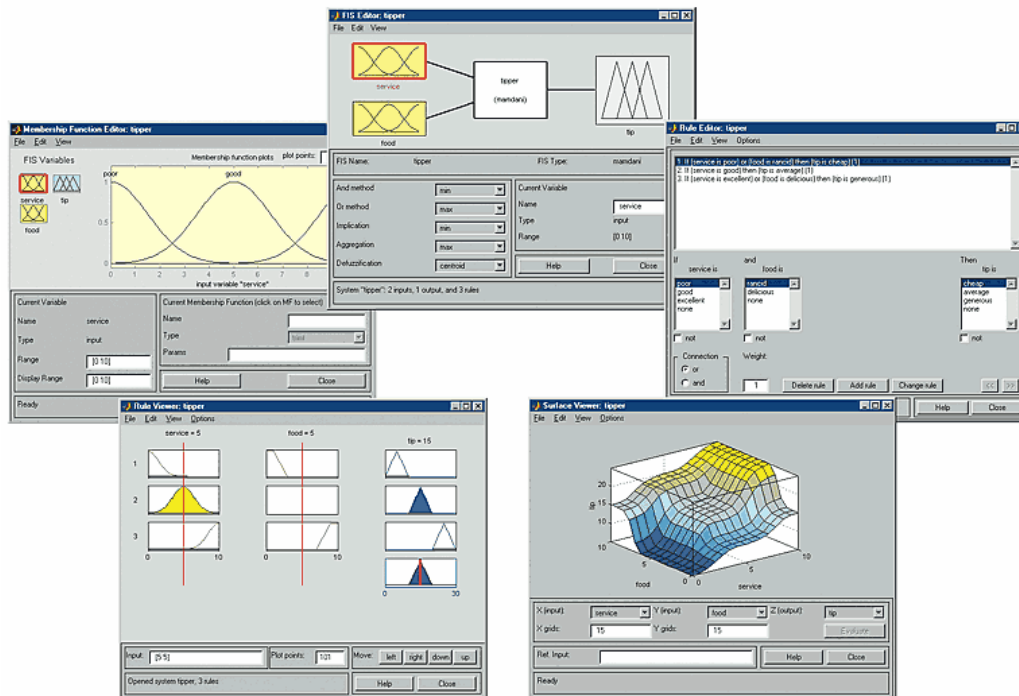


Fig 8.5: Fuzzy logic GUIs: The Membership Function Editor (top left), FIS Editor (center), Rule Editor (top right), Rule Viewer (bottom left), and Surface Viewer (bottom right). Click on image to see enlarged view.

8.4.3. Load the controller's parameters

Specify the controller's parameters in the m file "parameters_DAFc.m" (see section 8.3.1) and run it to load the controller's parameters to the workspace.

8.4.4. Perform simulation

Now, the model is ready for simulation. The simulation results are obtained by simply running the model and observing the results through the scopes. The controller can be tested with different parameters by repeating steps 8.4.2 to 8.4.4.

We have demonstrated the implementation process of DAFc of an inverted pendulum using Simulink. The process is quick and easy with 4 simple steps. A SSDAFc simulation of an inverted pendulum can be created by following the same steps. Fig 8.6 shows a simulink model of a SSDAFc of an inverted pendulum with the use of a high-gain observer.

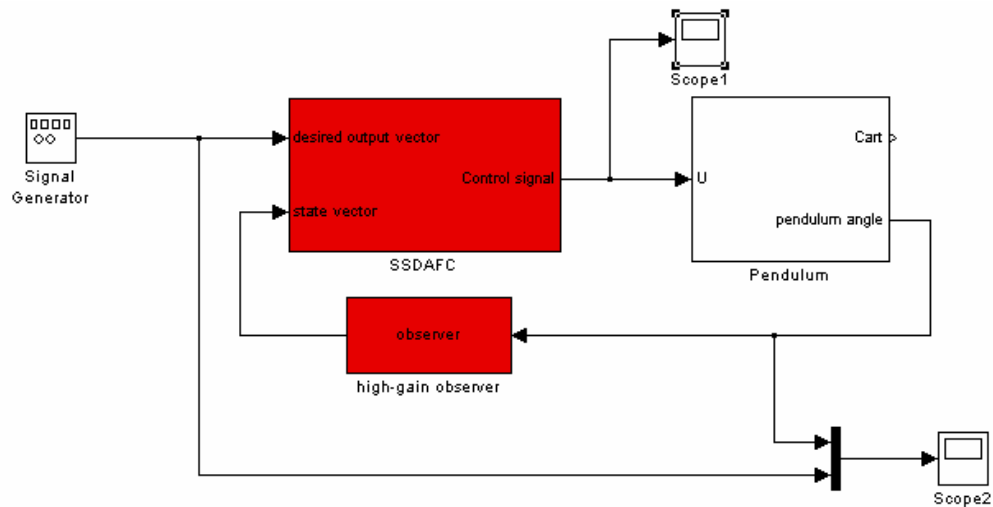


Fig 8.6 Observer-based SSDAFC of an inverted pendulum

The blocks in the AFC library not only can be used for simulation but also for real-time control. In the next section, we will show how a real-time control application can be easily set-up using our control simulink blocks.

8.5. Real-time AFC

Real-Time Windows Target is an add-on product of Simulink. It enables running of Simulink and Stateflow models in real time on a desktop or laptop PC for rapid prototyping or hardware-in-the-loop simulation of a control system. Creation, control, and real-time execution maybe done entirely through Simulink [98].

Real-Time Windows Target includes a set of I/O blocks that provide connections between the physical I/O board and real-time model. The real-time windows target library is shown in Fig 8.7.

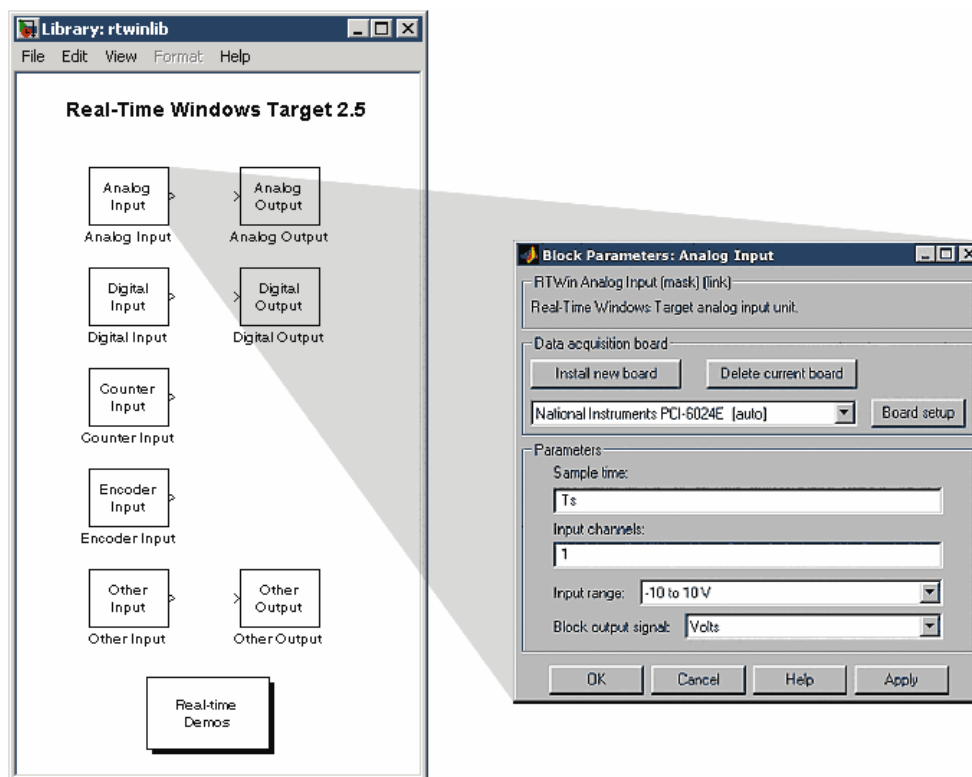


Fig 8.7 Real-time Windows Target library

By configuring these blocks with the physical I/O board, the adaptive fuzzy controller can be easily connected with the real physical plant. Fig 8.8 shows the setup of a real-time observer-based SSDAFC of an inverted pendulum.

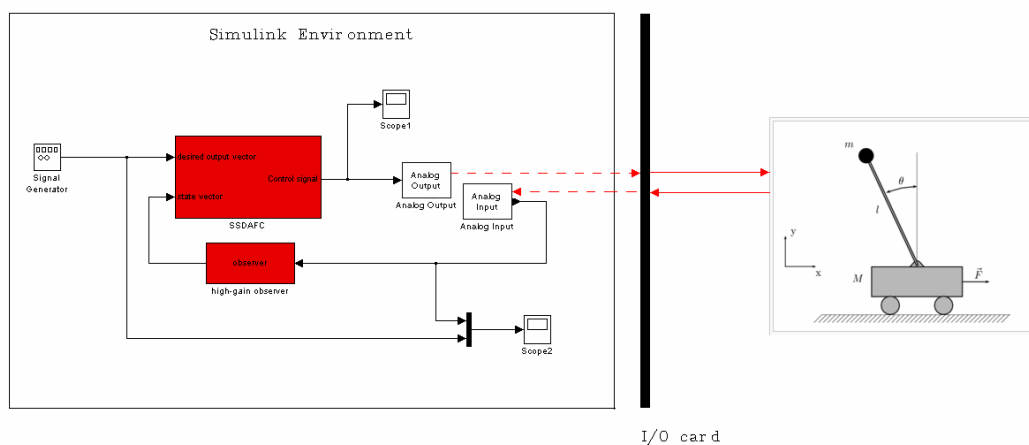


Fig 8.8 Real-time observer-based SSAFC of an inverted pendulum.

8.6. Conclusion

In this chapter, we have presented Matlab/Simulink implementation of our proposed control algorithms. A simulation application can be performed by only four simple steps. A simulation can be converted to a real-time control implementation by simply replacing the simulated plant by the I/O blocks provided in the real-time windows target library.

9. Chapter 9

DISCUSSION AND CONCLUSION

9.1. Discussion

9.1.1. Main contributions

In this research, an online SSAFC scheme has been developed. The main features of the proposed control scheme are:

- It is applicable for a number of different classes of continuous SISO nonlinear systems
- It needs less restriction on the controlled plants
- The stability of the overall system, especially when the structure changes, is guaranteed using the Lyapunov stability technique.
- The overall system is stable in the sense that all the variables are bounded (including number of rules generated) and the tracking error is uniformly ultimately bounded.
- For nonlinear systems in triangular forms, only one fuzzy system is needed (unlike the back-stepping approach where one fuzzy system is needed at each step).

The proposed control scheme makes practical application of AFC easier. Designers need to specify only a few design parameters and no longer have to specify the controller structure by trial and error. It saves the time and cost needed to check the extra restrictions on the controlled plants. It greatly reduces the complexity for nonlinear systems in triangular forms that are normally controlled using the back-stepping approach. It guarantees the stability of the system at any time and also guarantees that the fuzzy controller never exceeds the hardware capacity.

From the practitioners' point of view, the ability of the control scheme to control a wide range of classes of systems is a great advantage. When understanding the method, designers do not have to worry about choosing the right control configuration for a particular problem. This saves practitioners both learning time and designing time.

The Matlab and Simulink implementation of the controllers make simulations and real-time applications of AFC easy and fast. A simulation can be performed by

following four simple steps. Then, the simulation can be converted to a real-time application by simply re-connecting the controller to the real-time plant.

9.1.2. Limitations

The developed self-structuring algorithm suffers “the curse of dimensionality”. The number of added rules will dramatically increases for high-order systems. Thus, a much larger hardware capacity is needed for controlling high-order systems.

The control scheme guarantees that all the signals are bounded. However, similar to other AFC schemes, the bounds can be very conservative. Thus, information regarding these bounds is generally not useful for selecting design parameters. The control schemes in the literature suffer the same drawback. The reason is the bounds depend on the quantity $\|\underline{\theta}^*\|$ where $\underline{\theta}^*$ is the ideal adaptive parameter vector. $\|\underline{\theta}^*\|$ increases with the number of rules and can be arbitrarily large.

9.1.3. Future research

One future research direction would be to develop a self-structuring algorithm for high-order systems. Instead of using all possible rules when a membership function is added, we only add 1 rule at a time and all its corresponding memberships. The change would reduce the interpretability of the fuzzy system. This is a trade-off between computation and interpretability. However, as the system order increases, the fuzzy rules are harder to interpret anyway.

Another future research direction would be to develop tighter bounds. The popular way of proving the stability is to choose the Lyapunov function

$$V = \frac{1}{2} \underline{e}^T P \underline{e} + \frac{1}{2} (\underline{\theta} - \underline{\theta}^*)^T (\underline{\theta} - \underline{\theta}^*).$$

Thus, V depends on $\underline{\theta}^*$, and therefore the bounds depend on $\underline{\theta}^*$. If we choose a new Lyapunov function

$$V_{new} = \frac{1}{2} \underline{e}^T P \underline{e} + \frac{1}{2} (u - u^*)^2,$$

then V_{new} is much smaller than V . By investigating the approximation properties of derivatives of fuzzy systems, we may find a way to establish the stability for V_{new} .

Finally, a practical control problem generally includes selecting a suitable actuator, a reference model, and a controller. The relationship between actuator constraints, reference model, and AFC design parameters has not been thoroughly

investigated in the literature. It is of great practical interest to investigate how to incorporate actuator constraints and reference signal information into the choice of AFC design parameters. Understanding this would result in a more systematic design procedure.

9.2. Conclusion

In conclusion, the objectives set at the beginning of the research are met. The developed control scheme and implementation software make AFC easier. The results also open new research challenges with the ultimate purpose being to make AFC an easy-to-use control tool in practice.

Appendixes

Appendix 3.A

- Proof of 3.1(a.i) and 3.1(a.ii):

From the choice of the adaptive laws (3.3), (3.4), and (3.5), it is obvious that (a.i) and (a.ii) holds. Now, we are going to prove (a.iii).

- Proof of 1(a.iii)

Substituting control signal (3.2) to system (3.1), we have:

$$\dot{x}_n = y^{(n)} = f(\underline{x}) + g(\underline{x})u_C$$

Adding and subtracting $\hat{g}(\underline{x}|\underline{\theta}_g)u_C$ gives:

$$\begin{aligned} y^{(n)} &= (f(\underline{x}) - \hat{f}(\underline{x}|\underline{\theta}_f)) + (g(\underline{x}) - \hat{g}(\underline{x}|\underline{\theta}_g))u_C \\ &\quad + \underline{k}^T \underline{e} + r^{(n)} + \hat{g}(\underline{x}|\underline{\theta}_g)u_S \\ \Leftrightarrow e^{(n)} &= -\underline{k}^T \underline{e} + (\hat{f}(\underline{x}|\underline{\theta}_f) - f(\underline{x})) \\ &\quad + (\hat{g}(\underline{x}|\underline{\theta}_g) - g(\underline{x}))u_C - \hat{g}(\underline{x}|\underline{\theta}_g)u_S \end{aligned}$$

In the matrix form, we have:

$$\begin{aligned} \dot{\underline{e}} &= \Lambda_C \underline{e} \\ &\quad + \underline{b}_C \left[(\hat{f}(\underline{x}|\underline{\theta}_f) - f(\underline{x})) + (\hat{g}(\underline{x}|\underline{\theta}_g) - g(\underline{x}))u_C - \hat{g}(\underline{x}|\underline{\theta}_g)u_S \right] \end{aligned} \quad (3.A.1)$$

The Lyapunov design approach will be used to prove the stability of the system.

Consider the following Lyapunov function

$$V(t) = \frac{1}{2} \underline{e}^T P \underline{e} + \frac{1}{2\gamma_f} \underline{\phi}_f^T \underline{\phi}_f + \frac{1}{2\gamma_g} \underline{\phi}_g^T \underline{\phi}_g + \frac{1}{2\gamma_\omega} \psi_\omega^2 \quad (3.A.2)$$

in which $\underline{\phi}_f = \underline{\theta}_f - \underline{\theta}_f^*$, $\underline{\phi}_g = \underline{\theta}_g - \underline{\theta}_g^*$, and $\psi_\omega = W - \hat{W}$.

Its derivative along the solution (3.A.1) is

$$\dot{V} = \frac{\partial}{\partial t} \left(\frac{1}{2} \underline{e}^T P \underline{e} \right) + \frac{1}{\gamma_f} \underline{\phi}_f^T \dot{\underline{\phi}}_f + \frac{1}{\gamma_g} \underline{\phi}_g^T \dot{\underline{\phi}}_g + \frac{1}{\gamma_\omega} \psi_\omega \dot{\psi}_\omega \quad (3.A.3)$$

Using the facts that $(\Lambda_C \underline{e})^T = \underline{e}^T \Lambda_C^T$, $\underline{e}^T P \underline{b}_C = \underline{b}_C^T P \underline{e}$, and $\Lambda_C^T P + P \Lambda_C = -Q$, we have:

$$\frac{\partial}{\partial t} \left(\frac{1}{2} \underline{e}^T P \underline{e} \right) = -\frac{1}{2} \underline{e}^T Q \underline{e} + \underline{e}^T P \underline{b}_C \left[(\hat{f}(\underline{x}|\underline{\theta}_f) - f(\underline{x})) + (\hat{g}(\underline{x}|\underline{\theta}_g) - g(\underline{x}))u_C - \hat{g}(\underline{x}|\underline{\theta}_g)u_S \right]$$

Substituting to (3.A.3) gives

$$\begin{aligned}
\dot{V} &= -\frac{1}{2} \underline{e}^T Q \underline{e} \\
&+ \underline{e}^T P \underline{b}_C \left[\left(\hat{f}(\underline{x}|\underline{\theta}_f) - f(\underline{x}) \right) + \left(\hat{g}(\underline{x}|\underline{\theta}_g) - g(\underline{x}) \right) \mu_C - \hat{g}(\underline{x}|\underline{\theta}_g) u_S \right] \\
&+ \frac{1}{\gamma_f} \underline{\phi}_f^T \dot{\underline{\phi}}_f + \frac{1}{\gamma_g} \underline{\phi}_g^T \dot{\underline{\phi}}_g + \frac{1}{\gamma_\omega} \psi_\omega \dot{\psi}_\omega \\
&= -\frac{1}{2} \underline{e}^T Q \underline{e} \\
&+ \underline{e}^T P \underline{b}_C \left[\left(\hat{f}(\underline{x}|\underline{\theta}_f) - \hat{f}(\underline{x}|\underline{\theta}_f^*) \right) + \left(\hat{g}(\underline{x}|\underline{\theta}_g) - \hat{g}(\underline{x}|\underline{\theta}_g^*) \right) \mu_C \right] \\
&+ \frac{1}{\gamma_f} \underline{\phi}_f^T \dot{\underline{\phi}}_f + \frac{1}{\gamma_g} \underline{\phi}_g^T \dot{\underline{\phi}}_g + \frac{1}{\gamma_\omega} \psi_\omega \dot{\psi}_\omega \\
&+ \underline{e}^T P \underline{b}_C \left[\left(\hat{f}(\underline{x}|\underline{\theta}_f^*) - f(\underline{x}) \right) + \left(\hat{g}(\underline{x}|\underline{\theta}_g^*) - g(\underline{x}) \right) \mu_C \right] \\
&- \underline{e}^T P \underline{b}_C \hat{g}(\underline{x}) u_S
\end{aligned} \tag{3.A.4}$$

Now, considering the second and forth terms of (3.A.4), we have:

$$\begin{aligned}
&\underline{e}^T P \underline{b}_C \left(\hat{f}(\underline{x}|\underline{\theta}_f) - \hat{f}(\underline{x}|\underline{\theta}_f^*) \right) + \frac{1}{\gamma_f} \underline{\phi}_f^T \dot{\underline{\phi}}_f \\
&= \underline{e}^T P \underline{b}_C \underline{\phi}_f^T \zeta_f(\underline{x}) + \frac{1}{\gamma_f} \underline{\phi}_f^T \dot{\underline{\phi}}_f
\end{aligned}$$

Using the adaptive law (3.3), it is obvious that

$$\underline{e}^T P \underline{b}_C \underline{\phi}_f^T \zeta_f(\underline{x}) + \frac{1}{\gamma_f} \underline{\phi}_f^T \dot{\underline{\phi}}_f \leq 0 \tag{3.A.5}$$

Similarly, considering the third and fifth terms of (3.A.4), and the adaptive law (3.4), we have

$$\begin{aligned}
&\underline{e}^T P \underline{b}_C \left(\hat{g}(\underline{x}|\underline{\theta}_g) - \hat{g}(\underline{x}|\underline{\theta}_g^*) \right) + \frac{1}{\gamma_g} \underline{\phi}_g^T \dot{\underline{\phi}}_g \\
&= \underline{e}^T P \underline{b}_C \underline{\phi}_g^T \zeta_g(\underline{x}) + \frac{1}{\gamma_g} \underline{\phi}_g^T \dot{\underline{\phi}}_g \leq 0
\end{aligned} \tag{3.A.6}$$

Considering the last three terms of (3.A.4) and using the adaptive law (3.5) and

the fact that $|a|b - ab \tanh \frac{0.2785ab}{\varepsilon} \leq \varepsilon$, we have

$$\begin{aligned}
&\underline{e}^T P \underline{b}_C \left[\left(\hat{f}(\underline{x}|\underline{\theta}_f^*) - f(\underline{x}) \right) + \left(\hat{g}(\underline{x}|\underline{\theta}_g^*) - g(\underline{x}) \right) \mu_C \right] \\
&- \underline{e}^T P \underline{b}_C \hat{g}(\underline{x}) u_S + \frac{1}{\gamma_\omega} \psi_\omega \dot{\psi}_\omega \\
&\leq \begin{cases} \sigma_\omega (W - \hat{\omega})(\hat{\omega} - \omega_0) + \varepsilon & \text{if } (\hat{\omega} < \omega_{\max}) \\ \text{or } \left(\hat{\omega} = \omega_{\max} \text{ and } \left| \underline{e}^T P \underline{b}_C \right| \leq \sigma(\hat{\omega} - \omega_0) \right) \\ \left| \underline{e}^T P \underline{b}_C \right| (W - \hat{\omega}) + \varepsilon & \text{if } \left(\hat{\omega} = \omega_{\max} \text{ and } \left| \underline{e}^T P \underline{b}_C \right| > \sigma(\hat{\omega} - \omega_0) \right) \end{cases}
\end{aligned}$$

Moreover, we have that $|\underline{e}^T P \underline{b}_C| (W - \omega_{\max}) \leq 0$ and

$\sigma_\omega (W - \hat{\omega})(\hat{\omega} - \omega_0) \leq \frac{1}{4} \sigma_\omega (W - \omega_0)^2$. Therefore,

$$\begin{aligned} & \underline{e}^T P \underline{b}_C \left[\left(\hat{f}(x|\underline{\theta}_f^*) - f(x) \right) + \left(\hat{g}(x|\underline{\theta}_g^*) - g(x) \right) \mu_C \right] \\ & - \underline{e}^T P \underline{b}_C \hat{g}(x) u_S + \frac{1}{\gamma_\omega} \psi_\omega \dot{\psi}_\omega \\ & \leq \frac{1}{4} \sigma_\omega (W - \omega_0)^2 + \varepsilon \end{aligned} \quad (3.A.7)$$

From (3.A.4-3.A.7), we have

$$\dot{V} \leq -\frac{1}{2} \underline{e}^T Q \underline{e} + \frac{1}{4} \sigma_\omega (W - \omega_0)^2 + \varepsilon$$

$$\text{Let } d = \frac{1}{4} \sigma_\omega (W - \omega_0)^2 + \varepsilon. \quad (3.A.8)$$

From (3.A.8),

$$\begin{aligned} \dot{V} & \leq -\frac{1}{2} \underline{e}^T Q \underline{e} + d \\ & \leq -\frac{1}{2} \lambda_{\min}(Q) |\underline{e}|^2 + d \\ & \leq -\frac{1}{2} \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} \lambda_{\max}(P) |\underline{e}|^2 - \frac{1}{2\gamma_f} \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} \left(\left| \underline{\phi} \right|^2 - \max \left| \underline{\phi} \right|^2 \right) \\ & - \frac{1}{2\gamma_g} \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} \left(\left| \underline{\phi} \right|^2 - \max \left| \underline{\phi} \right|^2 \right) - \frac{1}{2\gamma_\omega} \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} (\psi_\omega^2 - \max(\psi_\omega^2)) + d \\ & \leq -k_{\min} \left(\frac{1}{2} \lambda_{\max}(P) |\underline{e}|^2 + \frac{1}{2\gamma_f} \left| \underline{\phi} \right|^2 + \frac{1}{2\gamma_g} \left| \underline{\phi} \right|^2 + \frac{1}{2\gamma_\omega} \psi_\omega^2 \right) + V_r \end{aligned}$$

$$\text{in which } \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} = k_{\min} \quad (3.A.9)$$

$$V_r = \frac{1}{2} \lambda_{\max}(P) \frac{2d}{\lambda_{\min}(Q)} + \frac{1}{2\gamma_f} \max \left| \underline{\phi} \right|^2 + \frac{1}{2\gamma_g} \max \left| \underline{\phi} \right|^2 + \frac{1}{2\gamma_\omega} \max(\psi_\omega^2) \quad (3.A.10)$$

And using the fact that $\frac{1}{2} \lambda_{\max}(P) |\underline{e}|^2 + \frac{1}{2\gamma_f} \left| \underline{\phi} \right|^2 + \frac{1}{2\gamma_g} \left| \underline{\phi} \right|^2 + \frac{1}{2\gamma_\omega} \psi_\omega^2 \geq V(t) \quad \forall t$, we

have

$$\begin{aligned} \dot{V} & \leq -k_{\min} V + k_{\min} V_r \\ \Rightarrow V(t) & \leq (V(0) - V_r) \exp(-k_{\min} t) + V_r \end{aligned} \quad (3.A.11)$$

Therefore,

$$V(t) \leq \max(V(0), V_r)$$

Since $V(t) \geq \frac{1}{2} \underline{e}^T P \underline{e} \geq \frac{1}{2} \lambda_{\min}(P) |\underline{e}|^2 \quad \forall t$, we have

$$\begin{aligned} \frac{1}{2} \lambda_{\min}(P) |\underline{e}|^2 &\leq \max(V(0), V_r) \\ \Leftrightarrow |\underline{e}| &\leq \sqrt{\frac{2 \max(V(0), V_r)}{\lambda_{\min}(P)}} \end{aligned}$$

This completes the proof of 3.1(a.iii)

- Proof of 3.1(a.iv)

Recall the definition of \underline{e} :

$$\begin{aligned} \underline{e} &= \underline{r} - \underline{y} = \underline{r} - \underline{x} \\ \Leftrightarrow \underline{x} &= \underline{r} - \underline{e} \end{aligned}$$

Therefore,

$$\begin{aligned} |\underline{x}| &\leq |\underline{r}| + |\underline{e}| \\ &\leq r_0 + \sqrt{\frac{2 \max(V(0), V_r)}{\lambda_{\min}(P)}} \end{aligned}$$

- Proof of 3.1(a.v)

From control (3.2),

$$\begin{aligned} |u| &\leq \frac{1}{|\hat{g}(\underline{x}|\underline{\theta}_g)|} \left(|\hat{f}(\underline{x}|\underline{\theta}_f)| + |r^{(n)}| + \|\underline{k}^T\| |\underline{e}| \right) \\ &+ \frac{1}{|\hat{g}(\underline{x}|\underline{\theta}_g)|} \left| \hat{\omega} \tanh \frac{0.2758 \underline{e}^T P \underline{b}_C \hat{\omega}}{\varepsilon} \right| \end{aligned}$$

It is clear that

$$\begin{aligned} |\hat{g}(\underline{x}|\underline{\theta}_g)| &\geq \hat{g}(\underline{x}|\underline{\theta}_g^L) \\ |\hat{f}(\underline{x})| &\leq \max(|\hat{f}(\underline{x}|\underline{\theta}_f^L)|, |\hat{f}(\underline{x}|\underline{\theta}_f^U)|) \\ |r^{(n)}| &\leq r_1 \\ |\underline{e}| &\leq \sqrt{\frac{2 \max(V(0), V_r)}{\lambda_{\min}(P)}} \\ \hat{\omega} &\leq \omega_{\max} \\ \left| \tanh \frac{0.2758 \underline{e}^T P \underline{b}_C \hat{\omega}}{\varepsilon} \right| &\leq 1 \end{aligned}$$

Thus, 3.1(a.v) holds.

- Proof of 3.1(b)

From (3.A.11), $V(t) \leq (V(0) - V_r) \exp(-k_{\min} t) + V_r$, we have

$$\lim_{t \rightarrow \infty} V(t) \leq V_r$$

Since $V(t) \geq \frac{1}{2} \underline{e}^T P \underline{e} \geq \frac{1}{2} \lambda_{\min}(P) |\underline{e}|^2 \geq \frac{1}{2} \lambda_{\min}(P) e^2$,

$$\begin{aligned} \lim_{t \rightarrow \infty} \frac{1}{2} \lambda_{\min}(P) e^2 &\leq V_r \\ \Leftrightarrow \lim_{t \rightarrow +\infty} |\underline{e}| &\leq \sqrt{\frac{2V_r}{\lambda_{\min}(P)}} \end{aligned}$$

This completes the proof of 3.1(b).

- Proof of 3.1(c)

From (3.A.8),

$$\dot{V} \leq -\frac{1}{2} \underline{e}^T Q \underline{e} + d$$

Using the fact that $\frac{1}{2} \underline{e}^T Q \underline{e} \geq \frac{1}{2} \lambda_{\min}(Q) |\underline{e}|^2 \geq \frac{1}{2} \lambda_{\min}(Q) e^2$, we have

$$\dot{V} \leq -\frac{1}{2} \lambda_{\min}(Q) e^2 + d$$

Rearranging terms and integrating, we have

$$\int e^2 dt \leq \int \left(-\frac{2\dot{V}}{\lambda_{\min}(Q)} + \frac{2d}{\lambda_{\min}(Q)} \right) dt$$

Since V is bounded we find that

$$\lim_{t \rightarrow \infty} \frac{1}{t} \int e^2 dt \leq \frac{2d}{\lambda_{\min}(Q)}$$

So the RMS error is bounded by

$$RMS = \sqrt{\lim_{t \rightarrow \infty} \frac{1}{t} \int e^2 dt} \leq \sqrt{\frac{2d}{\lambda_{\min}(Q)}}$$

Appendix 3.B

- Proof of theorem 3.2(a.i)

From (3.5), it is clear that theorem 3.2(a.i) holds.

- Proof of theorem 3.2(a.ii)

Consider the following Lyapunov function

$$V_1(t) = \frac{1}{2} \underline{e}^T P \underline{e} + \frac{1}{2\gamma_\omega} \psi_\omega^2 \tag{3.B.1}$$

The derivative of V_1 along the solution of (3.A.1) is

$$\begin{aligned}\dot{V}_1 = & -\frac{1}{2}\underline{e}^T Q \underline{e} + \underline{e}^T P \underline{b}_c \left[\left(\hat{f}(\underline{x}|\underline{\theta}_f^0) - f(\underline{x}) \right) + \left(\hat{g}(\underline{x}|\underline{\theta}_g^0) - g(\underline{x}) \right) \underline{u}_c \right] \\ & - \underline{e}^T P \underline{b}_c u_s + \frac{1}{\gamma_\omega} \psi_\omega \dot{\psi}_\omega\end{aligned}$$

Using assumption 3.4 and the same arguments in (3.A.7), we have

$$\begin{aligned}& \underline{e}^T P \underline{b}_c \left[\left(\hat{f}(\underline{x}|\underline{\theta}_f^0) - f(\underline{x}) \right) + \left(\hat{g}(\underline{x}|\underline{\theta}_g^0) - g(\underline{x}) \right) \underline{u}_c \right] - \underline{e}^T P \underline{b}_c u_s + \frac{1}{\gamma_\omega} \psi_\omega \dot{\psi}_\omega \\ & \leq \frac{1}{4} \sigma_\omega (W - \omega_0)^2 + \varepsilon = d\end{aligned}$$

Thus,

$$\dot{V}_1 \leq -\frac{1}{2} \underline{e}^T Q \underline{e} + d$$

Now, following the same procedure to prove theorem 3.1(a.iii), we can prove that theorem 3.2(a.ii) holds

$$|\underline{e}| \leq \sqrt{\frac{2 \max(V_1(0), V_{1r})}{\lambda_{\min}(P)}}$$

$$\text{where } V_{1r} = \frac{1}{2} \lambda_{\max}(P) \frac{2d}{\lambda_{\min}(Q)} + \frac{1}{2\gamma_\omega} \max(\psi_\omega)^2 \quad (3.B.2)$$

- Proof of theorem 3.2(a.iii) and theorem 3.2(a.iv):

Using the same arguments to proof theorem 3.1(a.iv) and theorem 3.1(a.v), we can show that theorem 3.2(a.iii) and theorem 3.2(a.iv) hold.

- Proof of theorem 3.2(b):

Similar to the proof of theorem 3.1(b), from $\dot{V}_1 \leq -\frac{1}{2} \underline{e}^T Q \underline{e} + d$, we can show that

$$\lim_{t \rightarrow +\infty} |\underline{e}| \leq \sqrt{\frac{2V_{1r}}{\lambda_{\min}(P)}}.$$

- Proof of theorem 3.2(c):

Using the same arguments as in the proof of theorem 3.1(c) and $\dot{V}_1 \leq -\frac{1}{2} \underline{e}^T Q \underline{e} + d$,

we can conclude that

$$RMS \leq \sqrt{\frac{2d}{\lambda_{\min}(Q)}}.$$

Appendix 4.B

Let $\underline{X}^j \in U_{\underline{X}}$. As $g(\underline{x})$, $\hat{u}(\underline{X}|\underline{\theta})$, and $u^*(\underline{X})$ are continuous at \underline{X}^j , for each $i=1\dots n, n+1$, there exists a $\delta_i^j > 0$ such that

$$\begin{aligned}
|X_i - X_i^j| < \delta_i^j \ (i=1 \dots n+1) &\Leftrightarrow \left| \left[g(\underline{X})u^*(\underline{X}) - g(\underline{X})\hat{u}(\underline{X}|\underline{\theta}) \right] - \left[g(\underline{X}^j)u^*(\underline{X}^j) - g(\underline{X}^j)\hat{u}(\underline{X}^j|\underline{\theta}) \right] \right| \leq \varepsilon^* \\
&\Leftrightarrow \left| \left[g(\underline{X})u^*(\underline{X}) - g(\underline{X})\hat{u}(\underline{X}|\underline{\theta}) \right] - g(\underline{X}^j)u^*(\underline{X}^j) - \hat{u}(\underline{X}^j|\underline{\theta}) \right| \leq \varepsilon^* \\
&\Leftrightarrow \left| \left[g(\underline{X})u^*(\underline{X}) - g(\underline{X})\hat{u}(\underline{X}|\underline{\theta}) \right] - c^j \left(u^*(\underline{X}^j) - \hat{u}(\underline{X}^j|\underline{\theta}) \right) \right| \leq \varepsilon^* \tag{4.A.1}
\end{aligned}$$

where $c^j = g(\underline{X}^j) > 0$.

Define

$$O_j = \left\{ \underline{X} \mid |X_i - X_i^j| \leq \delta_i^j \ (i=1 \dots n+1) \right\}$$

As $U_{\underline{X}}$ is compact, there exists a finite subfamily O_1, O_2, \dots, O_M such that

$$U_{\underline{X}} \subseteq O_1 \cup O_2 \cup \dots \cup O_M$$

Choose

$$A_i^j(X_i) = \alpha(X_i^j - \delta_i^j, X_i^j + \delta_i^j)(X_i), \ i=1 \dots n+1, \ j=1 \dots M \text{ such that}$$

$$\begin{cases} A_i^j(X_i^k) = 1 \text{ if } k = j \\ A_i^j(X_i^k) = 0 \text{ if } k \neq j \end{cases}, \ j, k = 1 \dots M \tag{4.A.2}$$

$$\theta_j^* = u^*(\underline{X}^j), \ j=1 \dots M \tag{4.A.3}$$

From (A.2),

$$\hat{u}(\underline{X}^j) = \sum_{k=1}^M \theta_k \zeta_k(\underline{X}^j) = \theta_1 \times 0 + \theta_2 \times 0 + \dots + \theta_j \times 1 + \dots + \theta_M \times 0 = \theta_j \tag{4.A.4}$$

Substituting (4.A.3) and (4.A.4) to (4.A.1), we have:

$$\left| \left[g(\underline{X})u^*(\underline{X}) - g(\underline{X})\hat{u}(\underline{X}|\underline{\theta}) \right] - c^j(\theta_j^* - \theta_j) \right| \leq \varepsilon^* \tag{4.A.5}$$

As $\zeta_j(\underline{X}) \neq 0$ for $\underline{X} \in O_j$ and $\zeta_j(\underline{X}) = 0$ for $\underline{X} \notin O_j$,

$$(A.5) \Rightarrow \left| \left[g(\underline{X})u^*(\underline{X}) - g(\underline{X})\hat{u}(\underline{X}|\underline{\theta}) \right] - c^j(\theta_j^* - \theta_j) \right| \zeta_j(\underline{X}) \leq \varepsilon^* \zeta_j(\underline{X})$$

Take the summation for $j=1 \dots M$,

$$\begin{aligned}
&\sum_{j=1}^M \left| \left[g(\underline{X})u^*(\underline{X}) - g(\underline{X})\hat{u}(\underline{X}|\underline{\theta}) \right] - c^j(\theta_j^* - \theta_j) \right| \zeta_j(\underline{X}) \leq \sum_{j=1}^M \varepsilon^* \zeta_j(\underline{X}) \\
&\Leftrightarrow \left| \left[g(\underline{X})u^*(\underline{X}) - g(\underline{X})\hat{u}(\underline{X}|\underline{\theta}) \right] \sum_{j=1}^M \zeta_j(\underline{X}) - \sum_{j=1}^M c^j(\theta_j^* - \theta_j) \zeta_j(\underline{X}) \right| \leq \varepsilon^* \sum_{j=1}^M \zeta_j(\underline{X})
\end{aligned}$$

Since $\sum_{j=1}^M \zeta_j(\underline{X}) = 1$, we have:

$$\left| \left[g(\underline{X})u^*(\underline{X}) - g(\underline{X})\hat{u}(\underline{X}|\underline{\theta}) \right] - \sum_{j=1}^M c^j(\theta_j^* - \theta_j) \zeta_j(\underline{X}) \right| \leq \varepsilon^* \tag{4.A.6}$$

Thus, $g(\underline{x})u^*(\underline{X}) - g(\underline{x})\hat{u}(\underline{X}|\underline{\theta})$ can be approximated by $\sum_{j=1}^M c^j (\theta_j^* - \theta_j) \zeta_j(\underline{X})$:

$$g(\underline{x})u^*(\underline{X}) - g(\underline{x})\hat{u}(\underline{X}|\underline{\theta}) = \sum_{j=1}^M c^j (\theta_j^* - \theta_j) \zeta_j(\underline{X}) + \varepsilon$$

where $|\varepsilon| \leq \varepsilon^*$ and c^j are some positive constants.

Appendix 6.A

Let $\underline{X}^j \in U_{\underline{X}}$. Since $f(\underline{x}, u(\underline{X}))$ is continuous with respect to $u(\underline{X})$, according to the Mean Value Theorem, there exists a positive constant $c^j > 0$ such that

$$f(\underline{x}^j, u^*(\underline{X}^j)) - f(\underline{x}^j, u(\underline{X}^j)) = c^j (u^*(\underline{X}^j) - u(\underline{X}^j)) \quad (6.A.1)$$

Since $f(\underline{x}^j, u^*(\underline{X}^j)) - f(\underline{x}^j, u(\underline{X}^j))$ is continuous at \underline{X}^j , for each $i = 1 \dots n, n+1$, there exists $\delta_i^j > 0$ such that

$$|X_i - X_i^j| < \delta_i^j \ (i = 1 \dots n+1) \Leftrightarrow [f(\underline{x}, u^*(\underline{X})) - f(\underline{x}, u(\underline{X}))] - [f(\underline{x}^j, u^*(\underline{X}^j)) - f(\underline{x}^j, u(\underline{X}^j))] \leq \varepsilon^* \quad (6.A.2)$$

Substituting (6.A.1) to (6.A.2) gives

$$|X_i - X_i^j| < \delta_i^j \ (i = 1 \dots n+1) \Leftrightarrow [f(\underline{x}, u^*(\underline{X})) - f(\underline{x}, u(\underline{X}))] - [c^j (u^*(\underline{X}^j) - u(\underline{X}^j))] \leq \varepsilon^* \quad (6.A.3)$$

where $c^j > 0$.

Define

$$O_j = \{\underline{X} \mid |X_i - X_i^j| \leq \delta_i^j \ (i = 1 \dots n+1)\}$$

As $U_{\underline{X}}$ is compact, there exist a finite subfamily O_1, O_2, \dots, O_M such that

$$U_{\underline{X}} \subseteq O_1 \cup O_2 \cup \dots \cup O_M$$

Choose

$$A_i^j(X_i) = \alpha(X_i^j - \delta_i^j, X_i^j + \delta_i^j)(X_i), \ i = 1 \dots n+1, \ j = 1 \dots M \text{ such that}$$

$$\begin{cases} A_i^j(X_i^k) = 1 \text{ if } k = j \\ A_i^j(X_i^k) = 0 \text{ if } k \neq j \end{cases}, \ j, k = 1 \dots M \quad (6.A.4)$$

$$\theta_j^* = u^*(\underline{X}^j), \ j = 1 \dots M \quad (6.A.5)$$

From (6.A.4),

$$u(\underline{X}^j) = \sum_{k=1}^M \theta_k \zeta_k(\underline{X}^j) = \theta_1 \times 0 + \theta_2 \times 0 + \dots + \theta_j \times 1 + \dots + \theta_M \times 0 = \theta_j \quad (6.A.6)$$

Substituting (6.A.5) and (6.A.6) to (6.A.3), we have:

$$|X_i - X_i^j| < \delta_i^j \ (i = 1 \dots n+1) \Leftrightarrow \left| \left[f(\underline{x}, u^*(\underline{X})) - f(\underline{x}, u(\underline{X})) \right] - \left[c^j (\theta_j^* - \theta_j) \right] \right| \leq \varepsilon^* \quad (6.A.7)$$

As $\zeta_j(\underline{X}) \neq 0$ for $\underline{X} \in O_j$ and $\zeta_j(\underline{X}) = 0$ for $\underline{X} \notin O_j$,

$$(6.A.7) \Rightarrow \left| \left[f(\underline{x}, u^*(\underline{X})) - f(\underline{x}, u(\underline{X})) \right] - c^j (\theta_j^* - \theta_j) \right| \zeta_j(\underline{X}) \leq \varepsilon^* \zeta_j(\underline{X})$$

Take the summation for $j = 1 \dots M$,

$$\begin{aligned} \sum_{j=1}^M \left| \left[f(\underline{x}, u^*(\underline{X})) - f(\underline{x}, u(\underline{X})) \right] - c^j (\theta_j^* - \theta_j) \right| \zeta_j(\underline{X}) &\leq \sum_{j=1}^M \varepsilon^* \zeta_j(\underline{X}) \\ \Leftrightarrow \left| \left[f(\underline{x}, u^*(\underline{X})) - f(\underline{x}, u(\underline{X})) \right] \sum_{j=1}^M \zeta_j(\underline{X}) - \sum_{j=1}^M c^j (\theta_j^* - \theta_j) \zeta_j(\underline{X}) \right| &\leq \varepsilon^* \sum_{j=1}^M \zeta_j(\underline{X}) \end{aligned}$$

Since $\sum_{j=1}^M \zeta_j(\underline{X}) = 1$, we have:

$$\left| \left[f(\underline{x}, u^*(\underline{X})) - f(\underline{x}, u(\underline{X})) \right] - \sum_{j=1}^M c^j (\theta_j^* - \theta_j) \zeta_j(\underline{X}) \right| \leq \varepsilon^* \quad (6.A.8)$$

Thus, $f(\underline{x}, u^*(\underline{X})) - f(\underline{x}, u(\underline{X}))$ can be approximated by $\sum_{j=1}^M c^j (\theta_j^* - \theta_j) \zeta_j(\underline{X})$:

$$f(\underline{x}, u^*(\underline{X})) - f(\underline{x}, u(\underline{X})) = \sum_{j=1}^M c^j (\theta_j^* - \theta_j) \zeta_j(\underline{X}) + \varepsilon$$

where $|\varepsilon| \leq \varepsilon^*$ and c^j are some positive constants.

References

- [1] L.-X. Wang, *Adaptive fuzzy systems and control: design and stability analysis*. New Jersey: Prentice-Hall: Englewood Cliffs, 1994.
- [2] S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable adaptive neural network control*. London: Kluwer Academic Publishers, 2002.
- [3] J. T. Spooner, M. Maggiore, R. Ordonez, and K. M. Passino, *Stable adaptive control and estimation for nonlinear systems: neural and fuzzy approximation techniques*. New York: John Wiley & Sons, 2002.
- [4] J. T. Spooner and K. M. Passino, "Stable adaptive control using fuzzy systems and neural networks," *Ieee Transactions on Fuzzy Systems*, vol. 4, pp. 339-359, 1996.
- [5] K. Fischle and D. Schroder, "An improved stable adaptive fuzzy control method," *Ieee Transactions on Fuzzy Systems*, vol. 7, pp. 27-40, 1999.
- [6] C. G. Moore and C. J. Harris, "Indirect Adaptive Fuzzy Control," *International Journal of Control*, vol. 56, pp. 441-468, 1992.
- [7] Y. W. Cho, C. W. Park, and M. Park, "An indirect model reference adaptive fuzzy control for SISO Takagi-Sugeno model," *Fuzzy Sets and Systems*, vol. 131, pp. 197-215, 2002.
- [8] C. W. Park and Y. W. Cho, "T-S model based indirect adaptive fuzzy control using online parameter estimation," *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 34, pp. 2293-2302, 2004.
- [9] C. W. Park and M. Park, "Adaptive parameter estimator based on T-S fuzzy models and its applications to indirect adaptive fuzzy control design," *Information Sciences*, vol. 159, pp. 125-139, 2004.
- [10] L. X. Wang, "Combining mathematical model and heuristics into controllers: An adaptive fuzzy control approach," *Fuzzy Sets and Systems*, vol. 89, pp. 151-156, 1997.
- [11] M. Hojati and S. Gazor, "Hybrid adaptive fuzzy identification and control of nonlinear systems," *Ieee Transactions on Fuzzy Systems*, vol. 10, pp. 198-210, 2002.
- [12] M. J. Er and S. H. Chin, "Hybrid adaptive fuzzy controllers of robot manipulators with bounds estimation," *Ieee Transactions on Industrial Electronics*, vol. 47, pp. 1151-1160, 2000.
- [13] C. H. Wang, T. C. Lin, T. T. Lee, and H. L. Liu, "Adaptive hybrid intelligent control for uncertain nonlinear dynamical systems," *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 32, pp. 583-597, 2002.
- [14] S. C. Tong and T. Y. Chai, "Fuzzy indirect adaptive control for a class of decentralized nonlinear systems," *International Journal of Systems Science*, vol. 29, pp. 149-157, 1998.
- [15] B. Yoo and W. Ham, "Adaptive fuzzy sliding mode control of nonlinear system," *Ieee Transactions on Fuzzy Systems*, vol. 6, pp. 315-321, 1998.
- [16] S. C. Tong, T. Wang, and J. T. Tang, "Fuzzy adaptive output tracking control of nonlinear systems," *Fuzzy Sets and Systems*, vol. 111, pp. 169-182, 2000.
- [17] Y. C. Chang, "Adaptive fuzzy-based tracking control for nonlinear SISO systems via VSS and H-infinity approaches," *Ieee Transactions on Fuzzy Systems*, vol. 9, pp. 278-292, 2001.

- [18] Y. X. Diao and K. M. Passino, "Stable fault-tolerant adaptive fuzzy/neural control for a turbine engine," *Ieee Transactions on Control Systems Technology*, vol. 9, pp. 494-509, 2001.
- [19] C. L. Hwang and C. Y. Kuo, "A stable adaptive fuzzy sliding-mode control for affine nonlinear systems with application to four-bar linkage systems," *Ieee Transactions on Fuzzy Systems*, vol. 9, pp. 238-252, 2001.
- [20] W. Y. Wang, Y. G. Leu, and C. C. Hsu, "Robust adaptive fuzzy-neural control of nonlinear dynamical systems using generalized projection update law and variable structure controller," *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 31, pp. 140-147, 2001.
- [21] J. H. Park, S. J. Seo, and G. T. Park, "Robust adaptive fuzzy controller for nonlinear system using estimation of bounds for approximation errors," *Fuzzy Sets and Systems*, vol. 133, pp. 19-36, 2003.
- [22] C. M. Lin and C. F. Hsu, "Adaptive fuzzy sliding-mode control for induction servomotor systems," *Ieee Transactions on Energy Conversion*, vol. 19, pp. 362-368, 2004.
- [23] Y. G. Leu, T. T. Lee, and W. Y. Wang, "Observer-based adaptive fuzzy-neural control for unknown nonlinear dynamical systems," *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 29, pp. 583-591, 1999.
- [24] A. Rubaai, "Direct adaptive fuzzy control design achieving H-infinity tracking for high performance servo drives," *Ieee Transactions on Energy Conversion*, vol. 14, pp. 1199-1208, 1999.
- [25] T. Y. Chai and S. C. Tong, "Fuzzy direct adaptive control for a class of nonlinear systems," *Fuzzy Sets and Systems*, vol. 103, pp. 379-387, 1999.
- [26] B. S. Chen, C. S. Wu, and Y. W. Jan, "Adaptive fuzzy mixed H-2/H-infinity, attitude control of spacecraft," *Ieee Transactions on Aerospace and Electronic Systems*, vol. 36, pp. 1343-1359, 2000.
- [27] W. Y. Wang, M. L. Chan, T. T. Lee, and C. H. Liu, "Adaptive fuzzy control for strict-feedback canonical nonlinear systems with H-infinity tracking performance," *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 30, pp. 878-885, 2000.
- [28] S. C. Tong and H. X. Li, "Direct adaptive fuzzy output tracking control of nonlinear systems," *Fuzzy Sets and Systems*, vol. 128, pp. 107-115, 2002.
- [29] W. Y. Wang, M. L. Chan, C. C. J. Hsu, and T. T. Lee, "H-infinity tracking-based sliding mode control for uncertain nonlinear systems via an adaptive fuzzy-neural approach," *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 32, pp. 483-492, 2002.
- [30] S. C. Tong, H. X. Li, and G. R. Chen, "Adaptive fuzzy decentralized control for a class of large-scale nonlinear systems," *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 34, pp. 770-775, 2004.
- [31] D. L. Tsay, H. Y. Chung, and C. J. Lee, "The adaptive control of nonlinear systems using the Sugeno-type of fuzzy logic," *Ieee Transactions on Fuzzy Systems*, vol. 7, pp. 225-229, 1999.
- [32] C. H. Wang, H. L. Liu, and T. C. Lin, "Direct adaptive fuzzy-neural control with state observer and supervisory controller for unknown nonlinear dynamical systems," *Ieee Transactions on Fuzzy Systems*, vol. 10, pp. 39-49, 2002.
- [33] J. H. Park and G. T. Park, "Robust adaptive fuzzy controller for non-affine nonlinear systems with dynamic rule activation," *International Journal of Robust and Nonlinear Control*, vol. 13, pp. 117-139, 2003.

- [34] H. B. Du, H. H. Shao, and P. J. Yao, "Adaptive neural network control for a class of low-triangular-structured nonlinear systems," *Ieee Transactions on Neural Networks*, vol. 17, pp. 509-514, 2006.
- [35] N. Essounbouli and A. Hamzaoui, "Direct and indirect robust adaptive fuzzy controllers for a class of nonlinear systems," *International Journal of Control Automation and Systems*, vol. 4, pp. 146-154, 2006.
- [36] J. H. Park, S. H. Huh, S. H. Kim, S. J. Seo, and G. T. Park, "Direct adaptive controller for nonaffine nonlinear systems using self-structuring neural networks," *Ieee Transactions on Neural Networks*, vol. 16, pp. 414-422, 2005.
- [37] J. H. Park, G. T. Park, S. H. Kim, and C. J. Moon, "Direct adaptive self-structuring fuzzy controller for nonaffine nonlinear system," *Fuzzy Sets and Systems*, vol. 153, pp. 429-445, 2005.
- [38] A. J. Calise, N. Hovakimyan, and M. Idan, "Adaptive output feedback control of nonlinear systems using neural networks," *Automatica*, vol. 37, pp. 1201-1211, 2001.
- [39] Y. H. Li, S. Qiang, X. Y. Zhuang, and O. Kaynak, "Robust and adaptive backstepping control for nonlinear systems using RBF neural networks," *Ieee Transactions on Neural Networks*, vol. 15, pp. 693-701, 2004.
- [40] T. Zhang, S. S. Ge, and C. C. Hang, "Adaptive neural network control for strict-feedback nonlinear systems using backstepping design," *Automatica*, vol. 36, pp. 1835-1846, 2000.
- [41] D. G. Chen and J. B. Yang, "Implementable adaptive backstepping neural control of uncertain strict-feedback nonlinear systems," *Advances in Neural Networks - Isnn 2006, Pt 2, Proceedings*, vol. 3972, pp. 875-880, 2006.
- [42] Y. S. Yang, G. Feng, and J. S. Ren, "A combined backstepping and small-gain approach to robust adaptive fuzzy control for strict-feedback nonlinear systems," *Ieee Transactions on Systems Man and Cybernetics Part a-Systems and Humans*, vol. 34, pp. 406-420, 2004.
- [43] D. Wang and J. Huang, "Neural network-based adaptive dynamic surface control for a class of uncertain nonlinear systems in strict-feedback form," *Ieee Transactions on Neural Networks*, vol. 16, pp. 195-202, 2005.
- [44] S. S. Ge, C. C. Hang, and T. Zhang, "Adaptive neural network control of nonlinear systems by state and output feedback," *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 29, pp. 818-828, 1999.
- [45] S. S. Ge, J. Zhang, and T. H. Lee, "Adaptive neural network control for a class of MIMO nonlinear systems with disturbances in discrete-time," *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 34, pp. 1630-1645, 2004.
- [46] J. Zhang, S. S. Ge, and T. H. Lee, "Output feedback control of a class of discrete MIMO nonlinear systems with triangular form inputs," *Ieee Transactions on Neural Networks*, vol. 16, pp. 1491-1503, 2005.
- [47] C. Wang, D. J. Hill, S. S. Ge, and G. R. Chen, "An ISS-modular approach for adaptive neural control of pure-feedback systems," *Automatica*, vol. 42, pp. 723-731, 2006.
- [48] T. P. Zhang, "Stable direct adaptive fuzzy control for a class of MIMO nonlinear systems," *International Journal of Systems Science*, vol. 34, pp. 375-388, 2003.
- [49] R. Ordóñez and K. M. Passino, "Stable multi-input multi-output adaptive fuzzy neural control," *Ieee Transactions on Fuzzy Systems*, vol. 7, pp. 345-353, 1999.

- [50] N. Golea, A. Golea, and K. Benmahammed, "Fuzzy model reference adaptive control," *Ieee Transactions on Fuzzy Systems*, vol. 10, pp. 436-444, 2002.
- [51] Y. Gao and M. J. Er, "Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems," *Fuzzy Systems, IEEE Transactions on*, vol. 11, pp. 462-477, 2003.
- [52] H. X. Li and S. C. Tong, "A hybrid adaptive fuzzy control for a class of nonlinear MIMO systems," *Ieee Transactions on Fuzzy Systems*, vol. 11, pp. 24-34, 2003.
- [53] S. C. Tong and H. X. Li, "Fuzzy adaptive sliding-mode control for MIMO nonlinear systems," *Ieee Transactions on Fuzzy Systems*, vol. 11, pp. 354-360, 2003.
- [54] F. L. Lewis, S. Jagannathan, and A. Yeşildirek, *Neural network control of robot manipulators and nonlinear systems*. London: Taylor & Francis, 1999.
- [55] S. S. Ge, T. H. Lee, and C. J. Harris, *Adaptive Neural Network Control of Robotic Manipulators*. London: World Scientific, 1998.
- [56] S. S. Ge, C. C. Hang, and T. Zhang, "Stable adaptive control for nonlinear multivariable systems with a triangular control structure," *Ieee Transactions on Automatic Control*, vol. 45, pp. 1221-1225, 2000.
- [57] S. S. Ge and C. Wang, "Adaptive neural control of uncertain MIMO nonlinear systems," *Ieee Transactions on Neural Networks*, vol. 15, pp. 674-692, 2004.
- [58] S. Seshagiri and H. K. Khalil, "Output feedback control of nonlinear systems using RBF neural networks," *Ieee Transactions on Neural Networks*, vol. 11, pp. 69-79, 2000.
- [59] F. Esfandiari and H. K. Khalil, "Output feedback stabilization of fully linearizable system," *Internation Journal of Control*, vol. 56, 1992.
- [60] M. Jankovic, "Adaptive output feedback control of nonlinear feedback linearizable system," *Internation Journal of Control*, vol. 10, pp. 1-18, 1996.
- [61] N. Hovakimyan, F. Nardi, and A. J. Calise, "A novel error observer-based adaptive output feedback approach for control of uncertain systems," *Ieee Transactions on Automatic Control*, vol. 47, pp. 1310-1314, 2002.
- [62] Y. G. Leu, W. Y. Wang, and T. T. Lee, "Observer-based direct adaptive fuzzy-neural control for nonaffine nonlinear systems," *Ieee Transactions on Neural Networks*, vol. 16, pp. 853-861, 2005.
- [63] N. Hovakimyan, A. J. Calise, and N. Kim, "Adaptive output feedback control of a class of multi-input multi-output systems using neural networks," *International Journal of Control*, vol. 77, pp. 1318-1329, 2004.
- [64] Y. C. Chang and H. M. Yen, "Adaptive output feedback tracking control for a class of uncertain nonlinear systems using neural networks," *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 35, pp. 1311-1316, 2005.
- [65] S. S. Ge and J. Zhang, "Neural-network control of nonaffine nonlinear system with zero dynamics by state output feedback," *Ieee Transactions on Neural Networks*, vol. 14, pp. 900-918, 2003.
- [66] N. Hovakimyan, F. Nardi, A. Calise, and N. Kim, "Adaptive output feedback control of uncertain nonlinear systems using single-hidden-layer neural networks," *Ieee Transactions on Neural Networks*, vol. 13, pp. 1420-1431, 2002.
- [67] S. Jagannathan, M. W. Vandegrift, and F. L. Lewis, "Adaptive fuzzy logic control of discrete-time dynamical systems," *Automatica*, vol. 36, pp. 229-241, 2000.

- [68] H. N. Nounou and K. M. Passino, "Stable Auto-Tuning of Adaptive Fuzzy/Neural Controllers for Nonlinear Discrete-Time Systems," *Fuzzy Systems, IEEE Transactions on*, vol. 12, pp. 70-83, 2004.
- [69] S. S. Ge, T. H. Lee, G. Y. Li, and J. Zhang, "Adaptive NN control for a class of discrete-time non-linear systems," *International Journal of Control*, vol. 76, pp. 334-354, 2003.
- [70] S. S. Ge, G. Y. Li, and T. H. Lee, "Adaptive NN control for a class of strict-feedback discrete-time nonlinear systems," *Automatica*, vol. 39, pp. 807-819, 2003.
- [71] S. S. Ge, G. Y. Li, J. Zhang, and T. H. Lee, "Direct adaptive control for a class of MIMO nonlinear systems using neural networks," *Ieee Transactions on Automatic Control*, vol. 49, pp. 2001-2006, 2004.
- [72] M. Alata, C. Y. Su, and K. Demirli, "Adaptive control of a class of nonlinear systems with a first-order parameterized Sugeno fuzzy approximator," *Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews*, vol. 31, pp. 410-419, 2001.
- [73] H. Han, C.-Y. Su, and Y. Stepanenko, "Adaptive control of a class of nonlinear systems with nonlinearly parameterized fuzzy approximators," *Fuzzy Systems, IEEE Transactions on*, vol. 9, pp. 315-323, 2001.
- [74] J. H. Park, G. T. Park, S. H. Kim, and C. J. Moon, "Output-feedback control of uncertain nonlinear systems using a self-structuring adaptive fuzzy observer," *Fuzzy Sets and Systems*, vol. 151, pp. 21-42, 2005.
- [75] L. H. Tsoukalas and R. E. Uhrig, *Fuzzy and Neural Approaches in Engineering*. New York: Willey, 1997.
- [76] F. O. Karray and C. D. Silva, *Soft Computing and Intelligent Systems Design: theory, tools, and applications*. New York: Pearson/Addison Wesley, 2004.
- [77] M. Negnevitsky, *Artificial Intelligence: a guide to intelligent systems*. Harlow, England: Addison Wesley, 2005.
- [78] M. F. Alexander Fink*, Oliver Nelles‡, Rolf Isermann*, "Supervision of nonlinear adaptive controllers based on fuzzy models," 2000.
- [79] E. Kim, "Output feedback tracking control of robot manipulators with model uncertainty via adaptive fuzzy logic," *Ieee Transactions on Fuzzy Systems*, vol. 12, pp. 368-378, 2004.
- [80] Y. X. Diao and K. M. Passino, "Adaptive neural/fuzzy control for interpolated nonlinear systems," *Ieee Transactions on Fuzzy Systems*, vol. 10, pp. 583-595, 2002.
- [81] Y. H. Park and G. T. Park, "Design of a robust adaptive fuzzy controller globally stabilizing the multi-input nonlinear system with state-dependent uncertainty," *Control and Cybernetics*, vol. 27, pp. 613-629, 1998.
- [82] F. C. Sun, Z. Q. Sun, Y. Y. Zhu, and W. J. Lu, "Stable neuro-adaptive control for robots with the upper bound estimation on the neural approximation errors," *Journal of Intelligent & Robotic Systems*, vol. 26, pp. 91-100, 1999.
- [83] F. C. Chen and C. C. Liu, "Adaptively Controlling Nonlinear Continuous-Time Systems Using Multilayer Neural Networks," *Ieee Transactions on Automatic Control*, vol. 39, pp. 1306-1310, 1994.
- [84] Y. S. Yang and J. S. Ren, "Adaptive fuzzy robust tracking controller design via small gain approach and its application," *Ieee Transactions on Fuzzy Systems*, vol. 11, pp. 783-795, 2003.

- [85] D. Velez-Diaz and Y. Tang, "Adaptive Robust Fuzzy Control of Nonlinear Systems," *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, vol. 34, pp. 1596-1601, 2004.
- [86] F. P. Da and W. Z. Song, "Fuzzy neural networks for direct adaptive control," *Ieee Transactions on Industrial Electronics*, vol. 50, pp. 507-513, 2003.
- [87] S. S. Ge and C. Wang, "Direct adaptive NN control of a class of nonlinear systems," *Ieee Transactions on Neural Networks*, vol. 13, pp. 214-221, 2002.
- [88] J. L. Castro, "Fuzzy-Logic Controllers Are Universal Approximators," *Ieee Transactions on Systems Man and Cybernetics*, vol. 25, pp. 629-635, 1995.
- [89] M. J. Er and S. Q. Wu, "A fast learning algorithm for parsimonious fuzzy neural systems," *Fuzzy Sets and Systems*, vol. 126, pp. 337-351, 2002.
- [90] F. J. Lin, C. H. Lin, and P. H. Shen, "Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive," *Ieee Transactions on Fuzzy Systems*, vol. 9, pp. 751-759, 2001.
- [91] C. F. Juang and C. T. Lin, "An on-line self-constructing neural fuzzy inference network and its applications," *Ieee Transactions on Fuzzy Systems*, vol. 6, pp. 12-32, 1998.
- [92] N. K. Kasabov and Q. Song, "DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *Ieee Transactions on Fuzzy Systems*, vol. 10, pp. 144-154, 2002.
- [93] P. P. Angelov and D. P. Filev, "An approach to Online identification of Takagi-Suigeno fuzzy models," *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 34, pp. 484-498, 2004.
- [94] M. F. Azeem, M. Hanmandlu, and N. Ahmad, "Structure identification of generalized adaptive neuro-fuzzy inference systems," *Ieee Transactions on Fuzzy Systems*, vol. 11, pp. 666-681, 2003.
- [95] J. H. Park and S. H. Kim, "Direct adaptive output-feedback fuzzy controller for a nonaffine nonlinear system," *Iee Proceedings-Control Theory and Applications*, vol. 151, pp. 65-72, 2004.
- [96] J. H. Park, S. J. Seo, D. W. Kim, and G. T. Park, "Adaptive fuzzy output-feedback controller for SISO affine nonlinear systems without state observer," in *Knowledge-Based Intelligent Information and Engineering Systems, Pt 4, Proceedings*, vol. 3684, *Lecture Notes in Artificial Intelligence*, 2005, pp. 549-558.
- [97] "<http://www.mathworks.com/products/simulink/>."
- [98] "<http://www.mathworks.com/products/rwt/>."